Ranking and reliable classification

Ranking and Reliable Classification

Ranking and Reliable Classification

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit Maastricht,
op gezag van de Rector Magnificus,
Prof. mr. G.P.M.F. Mols,
volgens het besluit van het College van Decanen,
in het openbaar te verdedigen
op woensdag 1 juli 2009, om 12:00 uur

door

Stijn Vanderlooy

Promotores: Prof. dr. H.J. van den Herik (Universiteit van Tilburg & Universiteit Leiden)
Prof. mr. Th.A. de Roos (Universiteit van Tilburg)
Prof. dr. rer. nat. E. Hüllermeier (Philipps-Universität Marburg)

Leden van de beoordelingscommissie:
Prof. dr. M. Gyssens (Universiteit Hasselt; voorzitter)
Prof. dr. A.P.A Broeders
Prof. dr. J. Fürnkranz (Technische Universität Darmstadt)
Prof. dr. J.C. Hage
Prof. dr. E.O. Postma (Universiteit van Tilburg)

The research has been funded by the Netherlands Organisation for Scientific Research (NWO), in the framework of the ToKeN project IPOL (grant number 634.000.435).

Dissertation Series No. 2009-21
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

ISBN 978-90-8559-537-3
Printed by Optima Grafische Communicatie, Rotterdam, The Netherlands.

Preface

Machine learning is an exciting interdisciplinary field that has been used successfully in many application domains. One of the main tools that machine learning provides are classifiers, used to generalise observations and make predictions in the (near) future. We want these classifiers to be as accurate as possible, and several techniques for improving their performance have been proposed. Nonetheless, in application domains where (among others) the consequences of incorrect predictions may be severe, the application of classifiers is limited so far. An example of such a domain with important implications for all of us is law enforcement. Here, it should be guaranteed that the classifiers make legally correct decisions. This is a difficult task and can be approached from various viewpoints and research directions.

In this thesis, owing to my interest in machine learning and the importance of applying classifiers in law enforcement, I will pursue several research directions in order to make the application of classifiers (more) safe and reliable. Actual implementation of the results of the research makes it possible for law enforcement to be more effective and efficient than is possible so far. More specifically, I mention the following three benefits: limited human resources and financial budgets are used in a more organised way, the number of incorrect predictions is reduced, and a better protection of the privacy rights of civilians is supported. Since the thesis has to address readers from two different fields, namely law and computer science, I have decided to provide the Chapters 3 to 6 with an own introduction and background section (they are named in a more specific way according to the topic). So, I hope to catch the attention of readers from both fields. Please note that, although the motivation for this thesis comes from law enforcement, the presented research is definitely also important for any other application domain.

The research presented in this thesis is the result of hard work, devotion to many interesting research topics, and a strong desire to learn as much as possible within a time period of four years. A few people strongly supported me in various ways and therefore they receive my sincerest acknowledgements. I would first like to thank Jaap van den Herik for giving me the opportunity to pursue a Ph.D. title at the Institute of Knowledge and Agent Technology (currently known as Department of Knowledge Engineering) of Maastricht University, The Netherlands. His enthusiasm supported my enthusiasm. Moreover, he considerably assisted me in improving my writing style. I am grateful to Theo de Roos and Joop Verbeek for introducing me in the law enforcement field. Ida Sprinkhuizen-Kuyper and Evgueni Smirnov

should also not be left unmentioned since they helped me with the development of the ROC isometrics approach. Thanks goes also to all my other colleagues who provided me with a working environment where there was room for pleasant discussions and laughter. My 2008 research visit to the University of Marburg, Germany, left an enormous positive impression on me. I would like to thank sincerely Eyke Hüllermeier, leader of the knowledge engineering and bio-informatics group, for learning me how to think about and perform research. Three chapters in this thesis and many more interesting findings were established under his supervision. It is difficult to express my gratitude to him in words. Sebastiaan Huntjens arranged almost all details concerning my stay in Marburg.

Besides the support that I received for the scientific part of my life, there are many people left to thank who created for me an enjoyable and heartwarming environment to live in. I would like to thank my caring family, fantastic neighbours, and crazy friends. Finally, I wish to extend my heartfelt gratitude to my parents and my sister for their continued support, joy, and love. This thesis is dedicated to them.

Stijn Vanderlooy, spring 2009.

– For my parents and sister

Acknowledgements

This research has been funded by the Netherlands Organisation for Scientific Research (NWO), in the framework of the ToKeN project IPOL (grant number 634.000.435). It has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems. Finally, I gratefully acknowledge financial support over the years by the Universiteitsfonds Limburg / SWOL.

Contents

Chapter 1

General Introduction

In recent years, we have seen an increased interest in the gathering of digital data of all kinds in the law enforcement field. These data need to be analysed in order to find new knowledge that can be used for fighting crime. Due to the enormous explosion of digital data, there is a need for new technologies to automate the analysis. From the artificial intelligence field, it is well-known that *machine learning* provides an effective and efficient way to analyse, understand, and learn from large data sets. However, applying machine learning in law enforcement is far from obvious.

In the thesis, we perform research in machine learning with special attention to the importance of applying classifiers in law enforcement. Our main goal is to present and analyse approaches that result in a safer and more reliable use of classifiers than is possible so far. This allows for a new form of law enforcement, called *intelligence led policing*, where the intelligence is gained by machine learning tools. Here, we remark that the research by itself is not limited to the law enforcement field.

This chapter presents the foundation for the research in the thesis. In Section 1.1 we provide background on intelligence led policing. In Section 1.2 we mention four applications of classifiers in law enforcement. Three specific problems that may arise are provided in Section 1.3. The problem statement that guides our research is formulated in Section 1.4, along with four research questions. Section 1.5 contains the research methodology. Finally, the structure of the thesis in given in Section 1.6.[1]

1.1 A New Form of Law Enforcement

It seems nowadays regular that newspapers contain articles reporting on criminal activities. The contents of these articles may vary from domestic violence and burglary to terrorist acts including suicide bombing and murder of media-covering personnel or innocent tourists. Not surprisingly, inhabitants of nations worldwide have a fee-

[1] This chapter is based on an article by Vanderlooy, Verbeek, and van den Herik (2007) and two conference publications (Vanderlooy, Verbeek, and van den Herik, 2005; Vanderlooy, Postma, Tuyls, and Sprinkhuizen-Kuyper, 2006a).

ling of unsafety. Some have this feeling when they are travelling in specific regions, others have it even in their home location. Although the feelings may differ by region and the actual percentage of inhabitants feeling unsafe may vary in time, it is well-known that the risk of becoming the victim of a crime has been increased globally (International Crime Victims Survey, 2008). So, preventing crime and providing security have become high-priority goals for nations worldwide.

To achieve these goals, governments follow two directions simultaneously. First, the legislation is adjusted in such a way that it provides more jurisdictions to law enforcement agencies and new (more severe) penalties are imposed on offenses. An example is to allow for more action in an early stage, e.g., suspicion about a crime has become sufficient for using phone taps. Also, new sentences are available to sanction offenders based on the length and severity of their criminal career. Details can be found in Blokland, Nagin, and Nieuwbeerta (2005b) and references therein. Second, the legislation is adjusted in such a way that it allows for more possibilities to gather, analyse, and exchange data about civilians (Kielman and Koelewijn 2005a; 2005b) An example is the recently proposed legislation update in the Netherlands and its implications to automated data processing, as discussed by Vanderlooy, Verbeek, and van den Herik (2005) who, in the end of their paper, propose several modifications for a safer and easier use of machine learning.

Owing to the second direction, it is possible to arrive at new knowledge and automated tools so that law enforcement can be executed adequately and in a timely fashion on a tactical, strategical, and operational level (Yen and Popp, 2005). This new form of law enforcement guided by means of data analysis is known as *intelligence led policing*. It has been conjectured to enable law enforcement agencies to solve crimes more efficiently than is possible so far, to prevent crimes, and to reduce risks of potential dangers to the society (Gill, 2000; Feeley and Simon, 2003; Cope, 2004; Tilley, 2005; De Hert, Huisman, and Vis, 2005; Thibault, Lynch, and McBride, 2006; Gill and Phythian, 2006; Ratcliffe, 2008).

In this thesis, we emphasize the enlarged possibility of using machine learning methods in law enforcement. We strongly believe that machine learning provides valuable and even indispensable tools for intelligence led policing, as we will motivate later by four well-chosen examples. In machine learning, a prominent concept is learning a mapping from instances (e.g., persons or bank accounts) to class labels (e.g., risk level) given a set of available example data (Mitchell, 1997; Hastie, Tibshirani, and Friedman, 2001; Alpaydin, 2004; Bishop, 2007). Such a mapping is called a *classifier* and it can be used to predict the class label of new, unseen instances. Several examples and formalisations of classifiers and related concepts will be provided at relevant places in the thesis.

For now, it is sufficient to know that classifiers receive a great deal of attention since integrating them in law enforcement practices is considered as an important prerequisite for intelligence led policing. The popularity of classifiers is further increased by the large public who is amazed by these "neat tools for solving crime". In fact, there is a popular television show called `Numb3rs` (Devlin and Lorden, 2007). A special federal agent receives assistance from his brother, a young but very gifted mathematics professor, who applies machine learning in order to solve various types

of crime cases. Admittedly, the show by itself is not realistic in the sense that the necessary data are always immediately available and cases are solved in a few days. However, the storylines come from true crime cases or are at least believable in the sense that the cases and proposed solutions may happen in reality.[2] Attention is also given to the reliability of the proposed solutions. The term *reliability* can take many forms and will play a central role throughout the thesis.

Despite several research papers that propose to apply classifiers in the field of law enforcement, the number of actual implementations of classifiers is quite limited in practice (Oskamp and Lauritsen, 2002). This is in particular the case when classifiers are used for difficult counterterrorism purposes (Jonas and Harper, 2006). Besides the complexity of the law enforcement field (Bibel, 2004), a reason is that the data to be analysed often include characteristics of civilians who may be innocent. Several normative questions arise, such as: (1) which data may be collected? (2) how reliable are the predictions of a classifier? and (3) what protections are to be maintained against potential privacy violations? After a thorough inspection of the literature, we believe that these and other related questions are often ignored or not dealt with adequately by researchers and practitioners working at the intersection of law enforcement and machine learning. So, we are interested in approaches to alleviate or possibly overcome a set of domain-specific problems and concerns in order to facilitate the implementation of intelligence led policing.

1.2 Example Applications

In this section, we mention four examples of classifiers in the law enforcement field that are currently used in practice or have been tested for implementation. Each example shows how a classifier can substantially improve the efficiency and effectiveness of law enforcement. The examples are: offender residence prediction (1.2.1), profiling (1.2.2), fraud detection (1.2.3), and predicting recidivism risk (1.2.4).

1.2.1 Offender Residence Prediction

It is difficult to predict human behaviour in the near future. As an illustration, we consider a serial criminal who has committed eight murders so far. Each murder is marked by a cross on a geographical map. Since the offender does not want to be arrested, the victims are assaulted and murdered at locations that appear randomly chosen. Hence, by inspecting the crosses on the map, we are assumed not to be able to predict the location of the next murder with high probability.

However, human behaviour is not completely random. There is always a pattern involved and this pattern can be learned by a classifier to predict a *hot-zone*, i.e., a small region in which the serial criminal's residence is located with high probability. This means that we work from the crosses backwards to the source of origin. Once the hot-zone is found, several methods (e.g., surveillance and DNA verification) can

[2] There is a book by Devlin and Lorden (2007) that accompanies the show and explains the methods and cases in more detail.

be used to find the offender among a group of persons with characteristics matching the evidence gathered so far. Hence, the classifier has become an investigating tool to help law enforcement agencies find the serial criminal.

Offender residence prediction is used in practice for some years now and several commercial packages including sophisticated classifiers and methods for further analysis are available for purchase (Rossmo, 1999; Goodwin, 2007). The first package, called `Rigel`, places a grid structure on top of a geographical map. A classifier is used to predict for each grid cell the probability that the offender has its residence there. The result is a map with a series of concentric coloured regions drawn, the colour depicting the learned probabilities (Rossmo, 1999). `Rigel` has several success stories, including catching Canada's most infamous serial killer, Clifford Robert Olsen. He attacked and murdered eleven children and he committed a number of sex assaults while he was doing this. Using the crime-location data of the encounter sites, the residence of the offender was accurately found. A second software package is `Predator`, which works in a similar fashion; see Goodwin (2007) for details.

1.2.2 Profiling

In their recent book, Devlin and Lorden (2007) write: "Machine learning, another branch of artificial intelligence, is perhaps the single most important tool within the law enforcement community's data-mining arsenal when it comes to profiling (and hence, one hopes, catching or preventing) criminals and terrorists".

A *profile* is a set of features that characterise the type of person under consideration. Features are searched for in large databases and a classifier is used that screens new persons in order to compute a suspicion score with respect to the learned profile. The actual classification depends on the value of this score. A high value generates a signal for a thorough (often manual) investigation.

The profile classifiers are daily used by law enforcement agencies and border guards, with the Computer Assisted Passenger Prescreening System (`CAPPS`) as the best example. From the attacks on September 11, 2001, it became clear that the screening of airline passengers should be done more carefully. This is a difficult task since over 693 million passengers in the USA alone pass through the airports annually (Airports Council International, 2008). Only a few of them may be actual risks for the society in the sense of being terrorists, drugs smugglers, and so on. Clearly, the number of security personnel is too limited for performing careful screening tasks and the waiting times at the airports should not increase too much. For these reasons, `CAPPS` was introduced as an automated system that uses profiles to signal potential terrorists for closer inspection.[3]

The system has been cancelled in 2004 because it received a great deal of criticism from privacy advocates due to the data that was used for profiling. Its design was also not bulletproof as shown by two mathematics students. In their paper, Chakrabarti and Strauss (2002) explain how terrorists can adapt their behaviour to render `CAPPS` less secure than systems that randomly select passengers for a detailed

[3] `CAPPS` was already used since 1999 but only after the 9-11 attacks the system was updated and considered as a true counterterrorism tool.

inspection. Hence, an unwanted backdoor for terrorists was provided. A successor is proposed, called `Secure Flight`, of which a full implementation is expected in 2010. At the moment of writing, Congress has blocked the system until the government proves that the system can pass ten tests for accuracy and privacy protection (Transportation Security Administration, 2008).

1.2.3 Fraud Detection

Fraud is a billion dollar business and has shown a large increase in the last decades due to advances in technology and communication (Jans, Lybaert, and Vanhoof, 2006). There exist various sorts of fraud, e.g., money laundering and telecommunications fraud. The early detection of credit card fraud is gaining popularity in law enforcement because the number of attempts for fraudulent transactions is increasing each year and a high level of crime activity is involved (Westphal, 2008).

As an illustration, we consider that the credit card company Barclaycard carries approximately 350 million transactions a year in the United Kingdom alone. Even when only 0.1% of these transactions are fraudulent for a value of 10 euros, the total loss for the company is 3.5 million euros. A fair number of these fraudulent transactions may be the work of a small but well-organised group of persons who, each day and on various locations throughout the country, make themselves richer with small amounts of money. In contrast to these transactions, there are of course also single fraudulent transactions that severely compromise the corresponding bank accounts. Thus, not surprisingly, there is large interest in applying machine learning classifiers for the rapid detection of fraud. Specific details are of course not made public, but a good general overview can be found in Bolton and Hand (2002).

1.2.4 Recidivism Risk

It is an empirical and widely-recognised fact that a small number of offenders are responsible for a disproportional part of all registered crimes. For example, Blokland *et al.* (2005b) study a group of offenders that are sentenced in a Dutch court in 1977. They show that only 8% of these offenders were responsible for 52% of the convictions of the complete group from 1977 to 2002. An offender who repeats criminal activities is called a *recidivist* and the act itself is called recidivism.

Recent political and social debates agree that predicting the recidivism risk of an offender has become an important tool for criminal penology and crime prevention (Blokland, 2005a). If an offender can be predicted to become a recidivist based on his past criminal career and life circumstances, then crime can be reduced significantly with relatively few juridical efforts. An important observation here is that behaviour in the past highly correlates with behaviour that still needs to take place, although external influences such as marriage and parenthood often distort the correlation severely (Blokland and Nieuwbeerta, 2005; Moffitt, 2006). Also, the available example data to learn a classifier is often limited and noisy in the sense that it only reports on registered crimes. Moreover, registrations may be incomplete or may contain mistakes. The task of predicting recidivism risk is far from trivial.

So far, the most comprehensive and largest study about predicting recidivism risk is by Blokland *et al.* (2005b). This study collected information of Dutch offenders involved in 4% of all cases that were either ruled upon by a judge or decided upon by the public prosecutor in 1977. For each offender involved in these cases, three types of information have been registered for which it was believed to be important factors for recidivism (Andrews, Bonta, and Wormith, 2006). First, information regarding the offense in 1977 was retrieved including severity and amount of public damage. Second, the criminal career until 2002 was unravelled in a similar fashion. Third, information on the life circumstances in 1977 such as marriage, occupation, and drugs addiction was collected. Only offenders for whom complete information was available were retained. This resulted in data regarding life circumstances and criminal careers of 4615 offenders. They were divided into four groups according to recidivism risk. We note that these groups were chosen in such a way that offenders in each group are very dissimilar from those in other groups. Hence, these groups might not be the best partitioning for use in practice, e.g., in practice it would be more relevant to have a group that contains the non-recidivists. Nonetheless, even with the given biased partitioning, experiments with a classifier showed that too many incorrect predictions are made. Our results (unpublished) with many other classifiers and different data preprocessing techniques led to the same conclusion. Even in the case when one focusses on specific offender populations by type (such as hard drugs addicts) or by region, the number of incorrect predictions is too large to be acceptable (Auerhahn, 1999; Schwalbe, Fraser, Day, and Cooley, 2006).

As we will see in the next section, incorrect predictions may lead to severe consequences and, hereby, the application of classifiers may become unjustified. We will also discuss two other domain-specific problems that can be traced back to the four example applications given above.

1.3 Three Domain-Specific Problems

In this section, we discuss three problems inherent to the field of law enforcement when classifiers are applied, namely: incorrect prediction costs (1.3.1), privacy violations (1.3.2), and dynamics of the class and cost distribution (1.3.3). We relate these problems to the aforementioned example applications. The key message is that conventional classifiers have difficulties to *guarantee* legally correct decisions.[4]

1.3.1 Incorrect Prediction Costs

For any non-trivial application it is reasonable to expect that classifiers make incorrect predictions. In law enforcement, the number of *allowed* mistakes should be set to allow for a justified trade-off between benefits associated with correct predictions and costs associated with incorrect predictions. The distinction between correct prediction and incorrect prediction can often only be made after human investigation.

[4]Legally correct decisions are defined as correct (ground truth) decisions for which the subsequent actions do not violate the law.

As an example, we consider the incorrect prediction costs in case of offender residence prediction. The cost of a false positive (i.e., a region incorrectly predicted as belonging to the hot-zone) leads to a waste of limited human resources since a follow-up investigation is done in the wrong region. The cost of a false negative (the hot-zone or a part of it is not detected) is a possible failure of the law enforcement task. However, in any case, the classifier provides at least some valuable information for the investigators: without it, the investigators would have no clear picture in which region(s) to focus their attention. Besides these costs, the benefit of a correct prediction can be considered as the capture of a serial criminal with much less manual efforts than possible otherwise. A similar reasoning can be given for the application of fraud detection and for profiling methods that conform with privacy regulations. So, the consequences of incorrect predictions are not seen as too severe (i.e., they are outweighted by the benefits of correct predictions) and therefore, these applications are used in real-life practice.

Identifying and quantifying the costs and benefits of predicting recidivism risk is more difficult. Blokland and Nieuwbeerta (2006) have studied the trade-off between costs and benefits using the "three strikes and you are out" policy that is used in many American states: a third conviction, independent of the severity or type of the offense, results automatically in a longer period of imprisonment. The rationale is that crime rates should be significantly reduced since the recidivists, who are very likely to be responsible for a large number of crimes still to be committed, are isolated from society. Four specific scenarios of this policy are considered: imprisonment lasts for twenty, ten, five, or two years. Using a set of example data, the study shows that already imprisonments of two years lead to a reduction of 33% in crime rates. However, the corresponding costs are also high and of various kinds. Longer imprisonments result in an explosion of the inmate population. Even an imprisonment of two years after the third conviction leads to a prison population that is six to seven times as high as without the policy. It results in large financial and material costs. Although it is likely that governments are prepared to cope with these costs (Lenhardt, 2006), the reliability of the predictions of the classifier is most concerning. Each prediction contains some intrinsic uncertainty since it cannot be guaranteed that the offenders sentenced to prison would have committed several new crimes if they were still in society (we remember the influence of external factors on criminal careers). So, a part of the offenders may be imprisoned for a longer time period than necessary, and this raises difficult ethical discussions, in particular since the classifier may predict incorrectly a high recidivism risk. Due to this reliability concern, the costs associated with predicting recidivism are considered to be even higher than the benefits. Thus, it is not surprisingly that so far one is not eager to use classifiers for predicting recidivism risk (Schwalbe *et al.*, 2006).

1.3.2 Privacy Violations

To learn a classifier, law enforcement agencies have access to a wide variety of data sets. The data sets may differ by type of information stored, by their location (regional, national, or international) and by access type (public or private). Large efforts

Table 1.1: A brief comparison between goal-oriented investigation and global-oriented investigation.

	Goal-Oriented	*Global-Oriented*
Goal	Specific knowledge	General knowledge
Cause	Strong suspicions or clues	No specific cause
Duration	Temporary	Permanent
Type of Data	Detailed data concerning the specific persons and/or events	Variety of data, often concerning persons not related to the specific events

are undertaken to migrate data sets with smart tools; see for example Kurlander (2005) and references therein. A main problem is that the data often contain sensitive information about civilians in general, e.g., race, religion, and other non-criminal features. Many privacy advocates are protesting and claim for more restrictions on the use of data mining and machine learning in such a way that the privacy rights of civilians are respected to predefined norms.

Below we will explain the problem into more detail using the difference between a goal-oriented investigation and a global-oriented investigation. The former is aimed to obtain knowledge for solving specific cases, while the latter is aimed to obtain any knowledge that leads to new investigations or contributes to the execution of current specific investigations. Table 1.1 summarizes the differences. A goal-oriented investigation takes place when there are strong suspicions or clues concerning a specific event, person, or small groups of events and/or persons. Hence, this type of investigation has a temporary character and it aims at improving a specific information position, e.g., to solve a murder case. The data to be analysed naturally consists of relevant facts, so disproportional privacy violations are not likely to occur. This is in contrast to the global-oriented investigation, which has a permanent character since it aims at improving and maintaining a general information position. It is not oriented toward a specific case or person, and therefore there are no strong suspicions and clues that can be followed. Consequently, the data to be analysed may contain (sensitive) facts about persons and events that are not related to known crimes. Privacy violations, possibly disproportional to the task at hand, can occur.

We may consider `CAPPS` as an example of an automated global-oriented investigation since the system is not targeted to identify specific terrorists (most of them are unknown anyway). Therefore, to learn its profile, it has to sift through data sets containing not only historical data and characteristics about terrorists, but also credit card transactions, telephone calls, driving history, and other information not registered for law enforcement purposes. It follows that it is challenging to protect the civilians' privacy right, although everyone agrees that without global-oriented investigation it is virtually impossible to enable intelligence led policing. Legislation should find an ideal balance between effective automatic data analysis and protection of the privacy of civilians (Vanderlooy *et al.*, 2007; Carter and Lansing, 2007).

So far, we have considered privacy violations due to the data that is analysed. However, it is important to note that incorrect predictions of a classifier can result in privacy violations as well. So, privacy violations can be seen as incorrect prediction costs. In CAPPS, the cost is small since the person that is flagged by the system has to step out a waiting queue for a detailed baggage check and for answering some default questions. This short hold-up is acceptable for everyone since it serves an important goal. However, in many other applications, the cost of privacy violations may be (too) high. For example, based on the predictions of a classifier, a surveillance can be started regarding a person who turns out to be innocent. As such, the surveillance was irrelevant for the task and a severe privacy violation has occurred.

In this thesis, we are interested in privacy violations as incorrect prediction costs and we will not focus on the data aspect (which data is allowed, how to ensure integrity, and related concerns and questions).

1.3.3 Dynamics of Class and Cost Distribution

A third domain-specific problem is the dynamics of the class distribution and cost distribution. The non-constant behaviour of these distributions is the result of the fact that crime trends and patterns change rapidly (McCue, 2003). This implies that classifiers should be easily adaptable to new *operating conditions*.

The class distribution describes the balance between positive examples and negative examples. It can change over time since crime is an evolving process. As an example, we consider a classifier that is able to flag fraudulent credit card transactions so that corresponding offenders are arrested. Then, for some time period, the fraudulent transaction rate may be lower as before until a new way to do money theft is found. This implies that the fraudulent transaction rate will increase again to high values since the classifier, so far, cannot detect the emerging pattern. In addition, it is often hard to (re-)learn a classifier when the number of positive instances is many times larger than the number of negatives, or vice versa. This is analogous to the method how we learn as humans: it is easier to understand something new when many examples of all kinds are given.

The cost distribution describes the balance between the costs of false positives and false negatives. This distribution varies with the societal and legal context. For example, consider the task of identifying upcoming terrorist attacks in the following two contexts. We will say that a "positive" instance is an upcoming terrorist attack. First, when terrorist threats are high, we may assume that agencies are pressed to work at (or cross) the boundaries of the civilians' privacy rights in order to ensure the public safety. Second, in contrast, when crime rates are low for a sustained time period, we may assume that securing privacy will become more important. The first context implies that the cost of not identifying an upcoming terrorist attack (i.e., the false negative cost) is substantially higher than that of violating the privacy of civilians (i.e., the false positive cost). Hence, the number of false negatives should be many times smaller than the number of false positives. The second context implies clearly a smaller difference between the error costs, and hereby, the difference between false positives and false negatives becomes less important.

1.4 Problem Statement and Research Questions

In the previous sections, we have seen that the number of (potential) applications of classifiers in law enforcement is large and the implications are important to all of us. However, so far there are difficult technical and ethical issues concerning the actual implementation of classifiers. Consequently, the problem statement to be discussed in this thesis reads as follows.

> **Problem statement**: To what extent can machine learning classifiers be used to increase the effectiveness and efficiency of law enforcement?

To find an answer to the problem statement, we performed research in three different but promising directions. These directions and the corresponding research questions are formulated in the subsections below.[5] In total, we will investigate four research questions in order to provide an answer to the problem statement.

1.4.1 Direction 1: Ranking Instances

We assume for a moment that a classifier can only predict two class labels, called positive class label and negative class label. For example, the classifier predicts by dichotomy whether a person is a terrorist. Since not all persons are easy to classify, it is beneficial to incorporate scores that reflect the confidence in the predictions.

Without loss of generality, we assume that the score increases with the likelihood that the instance under consideration belongs to the positive class. So, scores can be used for *ranking* instances from most likely positive to most likely negative. Despite the apparent simplicity of this approach, it has large benefits for practical applications, in particular in law enforcement. For example, focussing on the top of the list enables investigators to inspect the most likely cases and, hereby, the various costs of incorrect predictions can be reduced significantly.

It is not surprising that we aim at using the classifier which produces the best ranking of the instances that we will encounter in the future. Thus, the question becomes how to choose the best classifier among a set of candidates. We are in need of a performance metric to measure the ranking quality. In machine learning, the generally accepted metric for this purpose is the AUC (to be explained in Chapter 2). Roughly speaking, the AUC is high when many positive instances received a higher score than the negative instances. However, recently, more sophisticated variants of the AUC were proposed that also take into account the *absolute* difference in scores. These variants are motivated as being more reliable in the sense that they choose more often the best classifier from the candidate set. No theoretical justifications for this conjecture are given and empirical results are not so convincing. Therefore, our first research question reads as follows.

> **Research question 1 (RQ 1)**: To what extent is the AUC an effective performance metric for bipartite ranking when compared to its variants that consider the absolute values of the scores?

[5] For ease of exposition, we refrain from giving references in the next subsections. A full list of references can be found in the corresponding chapters.

Having established an answer to this research question, it is challenging to ask how to learn a classifier that maximises the AUC. Until now, not so much research has been done on AUC-maximising classifiers since most often the learning algorithms are designed to optimise other performance metrics such as accuracy. For the field of law enforcement, it is important that the predictions of the classifier can be understood, analysed, and interpreted. A decision tree allows for these requirements. So, the second research question reads as follows.

> **Research question 2 (RQ 2)**: How can the AUC of an interpretable and comprehensible classifier, such as decision trees, be optimised?

An approach that successfully deals with this research question is a valuable tool for law enforcement agencies and can in virtue be used for any task in which it is important to focus on the most reliable cases.

1.4.2 Direction 2: Preset Classification Performance

Despite the benefits of ranking over directly labelling instances, it is still unknown *a priori* how many mistakes the classifier makes (i.e., whether an instance with a high score is a true positive instance). In other words, only after a domain expert has investigated each case separately, it is known which cases were actually of interest.

Therefore, a different direction to the problem statement is to preset the classification performance of the classifier. So, we propose to apply classifiers only if they are able to *guarantee* a preset classification performance on each class. The values of these performances are chosen in such a way that the costs of incorrect classifications that still may occur are acceptable. For law enforcement, this means that we are often in need of an efficient and easy-to-use approach to improve performance until a desired level. We note that we speak here of improving (boosting) performance since it is natural to assume that the number of allowed mistakes is low and that the task is difficult to solve. Improving performance can be done by filtering out specific instances for which the classification is uncertain. Such instances are left unclassified for the moment, although they can be useful in their own right. Clearly, the number of unclassified instances should be as low as possible.

The benefit of such an approach is that, since performance is set prior to classification, we know what to expect from the classifier. The classifier becomes *reliable* and can be safely applied. We note that the approach should adapt easily to changes in the class and cost distribution. So, the third research question reads as follows.

> **Research question 3 (RQ 3)**: Can we develop a feasible approach by which a classifier is constructed that guarantees a preset classification performance on each class?

1.4.3 Direction 3: Multi-class Classification Problems

The two aforementioned research directions deal with a binary label space, i.e., instances are assigned one of two possible class labels. Although many problems can

be cast in this setting, there are also other problems which concern three or more class labels. These problems are called multi-class classification problems.

Admittedly, several state-of-the-art classifiers can only distinguish two disjoint classes of instances. A generally accepted strategy to solve a multi-class problem is to *decompose* it into a series of binary classification problems, e.g., we can consider each pair of classes as a separate subproblem. We learn a binary classifier for each subproblem. When provided with a new instance to be classified, we thus receive predictions from a set of (binary) classifiers. The question then becomes how to aggregate the set of predictions into a final classification of the instance.

Many of such *aggregation strategies* have been proposed but a solid theoretical foundation is missing. Therefore, the fourth research question reads as follows.

> **Research question 4 (RQ 4)**: Can we develop an aggregation strategy that is feasible in practice and shown to be optimal under reasonable conditions?

1.5 Research Methodology

We will address the problem statement and the four research questions by a proven research methodology. In essence, to answer each research question, we provide a theoretical analysis and an experimental verification. Below we present details.

To answer RQ 1, we will provide a theoretical analysis of the AUC and its three variants by introducing a unified framework in which the metrics can be analysed and compared. We present a bias-variance analysis and further investigate our results by searching for strong evidence from simulation studies. Moreover, extensive empirical experiments with several classifiers and benchmark data sets are performed.

The research dealing with RQ 2 is organised in the same way, except that we will first focus on related work to obtain a better understanding of some important previous results. Then, again a theoretical investigation, confirmed by experiments on synthetic data and real-life benchmark data sets, will be performed.

Our answer to RQ 3 will be based on identifying instances with uncertain classification. Instead of restricting the classifier to predict one of the possible class labels, the classifier is given the option to say "I do not know" and consequently it may refrain from classifying the instance (for which it is uncertain). The goal is then to design an approach for which the number of classification rejections is minimal, and yet, the desired performance values (as preset by a domain expert) are guaranteed. We introduce the ROC isometrics approach for this purpose and show in a formal way its validity for several performance metrics. Its benefits with respect to related approaches are discussed and we will test the approach in practice.

To answer RQ 4, we will develop an aggregation strategy that we call adaptive voting. We relate it to a widely-used and easy-to-understand aggregation strategy that is shown to be competitive in practice. Adaptive voting takes the strength of the binary classifiers into account. In this way, classifiers with a high performance are seen as more reliable than classifiers with a lower performance. So, the aggregation

strategy becomes less suspectible to (likely incorrect) outputs from the weak and unreliable classifiers. Although there is no guarantee on the classification performance, the proposed aggregation strategy is the best we can do. More specifically, we will show that our adaptive voting is optimal in the sense that it yields a maximum a posteriori (MAP) prediction of the class label of an instance. The conditions for this optimality are clearly stated and shown to be reasonable for actual implementation in practice. Simulation studies and experiments verify the usefulness of the strategy.

1.6 Structure of the Thesis

In this chapter we started with a motivation of our research from a viewpoint that argues for a safe and reliable use of machine learning in law enforcement. Then, we introduced the problem statement, the corresponding four research questions, and we explained our research methodology.

The remainder of the thesis is structured as follows. In Chapter 2 we provide general background information on classifiers and related topics. Having established notation and general concepts, in Chapter 3 we focus on the first research direction by providing a theoretical analysis and empirical verification of the AUC and its variants (to answer RQ 1). Subsequently, in Chapter 4 we continue with the first research direction and analyse how to render decision trees as good rankers (to answer RQ 2). In Chapter 5 we focus on the second research direction and present an approach for constructing reliable classifiers (to answer RQ 3). Thereafter, we pursue the third research direction dealing with the multi-class classification setting in Chapter 6, where we introduce and analyse our optimal aggregation strategy (to answer RQ 4). For the Chapters 3 to 6, we note that the corresponding research methodology is naturally reflected in the chapter structure. Finally, in Chapter 7 we first answer the four research questions and then come to a concluding answer to the problem statement. We end the thesis with recommendations for future research.

Chapter 2

General Background

The focus of the present research is on the use of machine learning techniques to improve the effectiveness and efficiency of law enforcement agencies. This chapter provides background information in support of the research. After a short introduction in Section 2.1, we explain in Sections 2.2 to 2.4 the three machine learning settings that we use throughout the thesis: (1) binary classification learning and its extension to bipartite ranking, (2) multi-class classification learning, and (3) label ranking. We provide a summary of the chapter in Section 2.5.

2.1 Supervised Machine Learning

A fundamental concept in machine learning is that of *supervised learning*. In its most basic form, the objective is to "learn" a mapping from a given input space to a corresponding space of class labels. For example, the instance space can be all possible descriptions of credit card transactions, while the label space contains two elements to identify transactions as fraudulent and non-fraudulent. The individual features that compose a description are often identified by a domain expert and recorded by an automated system. Clearly, the quality of the features has a large impact on the quality of the mapping that has to be learned.

Formally, an input is considered as any instance from an input space $\mathcal{X}$. We do not make assumptions on this space. For example, the instance space can be real-numbered, i.e., $\mathcal{X} = \mathbb{R}^d$ with d the number of features, or it can be more complex in the sense that the features are of different types such as real numbers and binary variables. Even graphs (e.g., graphs encoding links between terrorists) can be considered as instances. The class label space is denoted by $\mathcal{L}$.

Definition 2.1. *A (discrete) classifier is a mapping $f : \mathcal{X} \to \mathcal{L}$ with $\mathcal{X}$ the instance space and $\mathcal{L}$ the class label space.*

Sometimes a classifier is called a *model*. In the classifier (model) the discriminative characteristics of the features are described in such a way that the classifier

can decide on the class label of instances. Learning of the classifier f is done by detecting patterns inherent in a set of available examples. An *example* is defined as an input together with its class label. So, we have that an example is of the form $z = (\boldsymbol{x}, \lambda_{\boldsymbol{x}})$ where $\boldsymbol{x} \in \mathcal{X}$ is the instance and $\lambda_{\boldsymbol{x}} \in \mathcal{L}$ is its class label. The available examples to learn from are called training data and denoted by

$$S = \{(\boldsymbol{x}_1, \lambda_{\boldsymbol{x}_1}), \ldots, (\boldsymbol{x}_s, \lambda_{\boldsymbol{x}_s})\} = \{z_1, \ldots, z_s\} \subset \mathcal{X} \times \mathcal{L} \ , \tag{2.1}$$

where we assume that each example is independently generated by the same unknown probability distribution $\mathbb{P}$ over the example space $\mathcal{Z} = \mathcal{X} \times \mathcal{L}$. This is the *independent and identically distributed (iid) assumption*, which is considered in most work about machine learning. It is important to note that we do not assume a deterministic mapping from instances to class labels. Instead, for each instance, there is a probability for each of the class labels. Often the class label with highest probability (i.e., the most likely one) should be predicted.

The algorithm that learns a classifier is called the *learning algorithm*. Throughout this thesis, we will encounter many examples of such algorithms. We refrain from presenting an overview here. Different algorithms often search in the training data for different types of patterns that can be exploited. By validating the quality of the detected patterns, we can expect to have learned something useful so that we can generalise over the examples. Thus, the learned classifier can be used to predict the class labels for new, unseen instances. The main goal is, of course, to learn a mapping that is as accurate as possible (Mitchell, 1997; Hastie *et al.*, 2001; Alpaydin, 2004; Bishop, 2007).

Depending on the label space $\mathcal{L}$, several settings of supervised machine learning can be distinguished. We consider two settings, namely the classification learning setting and the label ranking setting. In the *classification learning* setting, the label space is restricted to an unordered set of finite cardinality, i.e., $\mathcal{L} = \{\lambda_1, \ldots, \lambda_m\}$. We have a binary classification problem when $m = 2$. In this case, we will write $\mathcal{L} = \{+1, -1\}$ and accordingly say that an instance is either positive or negative. Learning in the case of a label space with more than two elements ($m > 2$) is referred to as multi-class learning. The *label ranking* setting assumes a total order over the class labels. The label space is then the set of all permutations of the class labels. An ordering can be interpreted as a preference relation, which is a natural interpretation (or even requirement) in many application domains. In the next sections, we will explain these learning settings in more detail.

Independent of the specific characteristics of the label space, the quality of the learned classifier $f : \mathcal{X} \to \mathcal{L}$ has to be assessed by a *performance metric*. A large number of these metrics have been proposed and we will use many of them in the thesis. For now, it is sufficient to provide a general framework to assess the quality of the classifier. More specifically, a classifier needs to minimise the expected risk

$$R(f) = \int_{\mathcal{X} \times \mathcal{L}} l(f(\boldsymbol{x}), \lambda_{\boldsymbol{x}}) \, d\mathbb{P}(\boldsymbol{x}, \lambda_{\boldsymbol{x}}) \ , \tag{2.2}$$

where $l : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ is a loss function that measures the discrepancy between the

prediction of the classifier and the correct class label.[1] The loss function should be chosen dependent on the specific problem and learning setting; it defines the performance metric to be used. A popular choice in the classification learning setting is the 0-1 loss function which outputs 0 if $f(\boldsymbol{x}) = \lambda_{\boldsymbol{x}}$ and 1 otherwise. So, we punish the classifier whenever it makes an incorrect prediction. This leads to the error rate as performance metric. Several other loss functions will be introduced in the thesis when necessary. Of course, the expected risk often cannot be explicitly computed. What we can do, however, is to compute an estimate of it using a finite test set that is independent of the training set. The integral in (2.2) is then transformed into a sum over a set of test instances. Clearly, the larger and more representative the test set, the more reliable the estimate of the expected risk is.

2.2 Binary Classification and Bipartite Ranking

Many problems can be formulated as binary classification tasks. We have mentioned four examples in the field of law enforcement in Chapter 1. Below, we discuss discrete binary classifiers (2.2.1), scoring classifiers (2.2.2), and we give more details about how to summarise and visualise the performance of these classifiers (2.2.3).

2.2.1 Discrete Classifier for Classification

A discrete binary classifier maps instances to class labels, i.e., $f : \mathcal{X} \to \{+1, -1\}$. For example, a classifier might predict if credit card transactions (instances) are fraudulent or not fraudulent (class labels). A correctly classified positive instance is called a true positive. If the positive instance is classified as negative, then it is a false negative. A true negative and false positive are defined analogously.

The total number of true positives, false positives, true negatives, and false negatives are denoted by TP, FP, TN, and FN, respectively. Henceforth, we will refer to them as the *performance statistics*. In addition, we write the number of positive instances as $P = TP + FN$ and the number of negative instances as $N = TN + FP$. From these numbers we derive the following four rates:

$$\begin{array}{ll} tpr = \dfrac{TP}{TP + FN} & \qquad tnr = \dfrac{TN}{TN + FP} \\ \text{and} & \\ fpr = \dfrac{FP}{FP + TN} & \qquad fnr = \dfrac{FN}{TP + FN} \end{array} , \tag{2.3}$$

where true positive rate is denoted by tpr and true negative rate by tnr.[2] False positive rate and false negative rate are denoted by $fpr = 1 - tnr$ and $fnr = 1 - tpr$, respectively. Many performance metrics can be written in terms of these four rates.

[1] We note that the expected risk (2.2) is explicitly written for the classification setting. In the label ranking setting, the loss function may have label rankings as arguments.

[2] Depending on the research field tpr is sometimes called positive accuracy, recall, or sensitivity. Negative accuracy and specificity are synonyms for tnr.

2.2.2 Scoring Classifier for Bipartite Ranking

In contrast to a discrete classifier, a scoring classifier is an $\mathcal{X} \to \mathbb{R}$ mapping. Without loss of generality, by possibly applying a linear transformation, we can assume the outputs of the scoring classifier to lie in the unit interval.

Definition 2.2. *A scoring classifier is a mapping $f : \mathcal{X} \to [0, 1]$ and $f(\boldsymbol{x})$ is said to be the score of the instance $\boldsymbol{x} \in \mathcal{X}$.*

The score of an instance is interpreted as the probability or, more generally, as a degree of confidence that the instance belongs to the positive class. So, the closer the output of a scoring classifier to 1, the more confident it is that the instance is a positive one. Values close to 0 indicate that the instances are likely to be negatives.

Many of the example applications in the previous chapter employ scoring classifiers, such as: predict for a small region the likelihood that an offender lives there, and predict the recidivism risk. Instead of relying on scoring classifiers, one could also rely on the crisp classifications of a discrete classifier. However, some information is then lost, namely: scores reflect the uncertainty in the classifications (for some instances it is more difficult to predict the class label than for other instances). Also, scores can be used to obtain an ordering on the instance space and domain experts can focus on specific parts of interest (e.g., the most likely positives have to be manually investigated or additional postprocessing is needed for difficult-to-classify instances). When the goal is to rank all positives higher than all negatives, then the task is known as *bipartite ranking*.

Scores are provided (explicitly or implicitly) by any classifier and therefore can be assumed to be always available. Applying a numerical threshold on scores transforms a scoring classifier into a discrete classifier, i.e., the bipartite ranking problem is reduced to a simpler binary classification problem. An instance is then classified as positive if its score is higher than or equal to the chosen threshold, and otherwise as negative. We note that a scoring classifier is almost never optimal, i.e., there will exist negative instances that received a higher score than some positive instances. In other words, the produced scores do not allow to distinguish correctly between the positive class and the negative class. Therefore, applying a threshold often results in (sometimes unaffordable) incorrect classifications.

An illustration is given in Fig. 2.1. When, at a certain score on the horizontal axis, the density of a class is non-negative, then the score can be assigned to an instance of that class. Thus, the overlap between the class probability density functions of the scores represents the non-optimality of the scoring classifier, since scores in the overlap can be assigned to both positive and negative instances (albeit with varying probability). A possible threshold is indicated by the straight vertical line and the relation with the four performance statistics (2.3) is also presented.

Clearly, the value of the threshold determines the trade-off between the number of false positives and false negatives. A smaller threshold value decreases FN and increases TP. Simultaneously, however, the smaller threshold also decreases TN and increases FP. A larger threshold value results in the opposite movement of the four performance statistics. In the following subsection, we use the concept of the ROC curve to visualise this trade-off such that a suitable threshold is found.

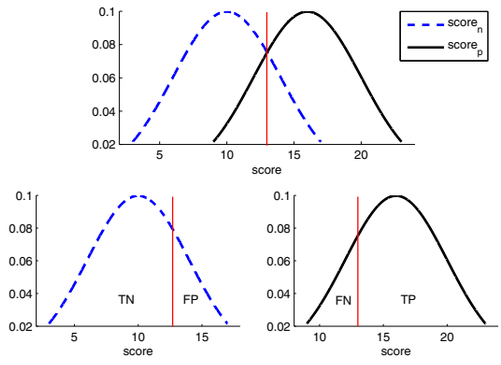


Figure 2.1: Score density functions of the negative class and positive class: (above) a possible threshold is indicated by the vertical line, and (below): the relation between the threshold and the performance statistics is given by the four distinct areas.

2.2.3 ROC Analysis for Visualising Performance

The *receiver operating characteristic (ROC) analysis* originated from signal detection theory (Swets, 1964). Recently, it was shown to be convenient for machine learning; see for example Swets, Dawes, and Monahan (2000), Provost and Fawcett (2001), Fawcett (2003) and references therein. Below we discuss its three main concepts, namely: ROC space, ROC curve, and area under the ROC curve.

The ROC Space

The ROC space is a convenient way to visualise and compare discrete classifiers in terms of their performance statistics as defined by (2.3).

Definition 2.3. *The ROC space is a two-dimensional space with fpr on the horizontal axis and tpr on the vertical axis. A discrete classifier is a point in this space.*

Figure 2.2 shows five discrete classifiers applied to the same set of test instances. The location of a point in ROC space gives an interpretation to the corresponding classifier. For example, the points $(0,0)$ and $(1,1)$ represent classifiers that always predict the negative class and positive class, respectively. An optimal classifier corresponds to point $(0,1)$, so the classifier labelled D is optimal. Point $(1,0)$ represents the classifier that makes only incorrect classifications. The ascending diagonal $(0,0) - (1,1)$ represents the strategy of random classification: any point (a,a) is obtained by predicting the positive class with probability a and the negative class with probability $1-a$. So, classifier C predicts the positive class 70% of the time. It can be expected to obtain 70% of the positives correct, but its false positive rate will increase to 70% as well. Thus, it is a random classifier. The classifier E is performing worse than random since it lies in the lower right triangle of ROC space.

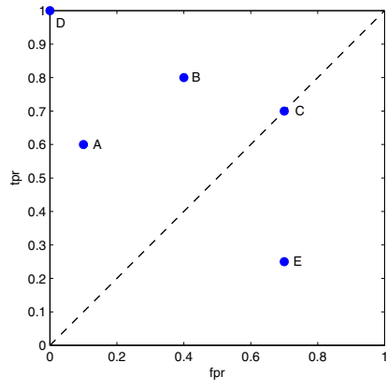
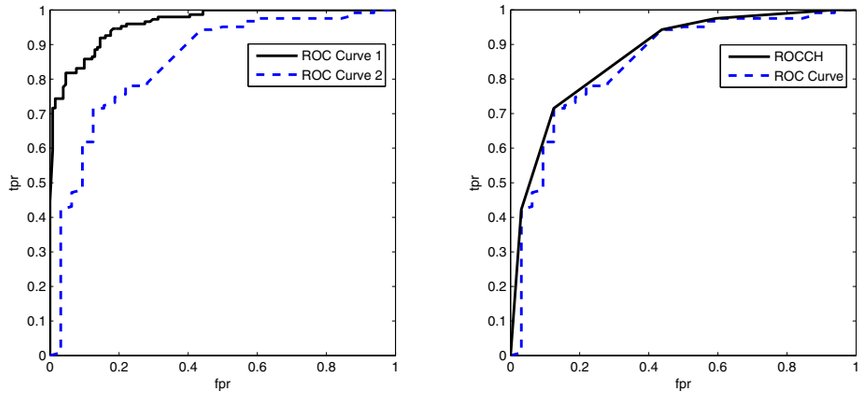


Figure 2.2: The ROC space with five discrete classifiers (A, B, C, D, E) evaluated on the same set of instances. The performances of these five classifiers can be assessed by comparing the locations of the corresponding points in ROC space.



(a) (b)

Figure 2.3: An ROC curve shows the performance of a scoring classifier: (a) curve 1 dominates curve 2, and (b) a curve and its ROCCH. Note that an ROC curve is always dominated by its convex hull.

The ROC Curve

A discrete classifier leads to a single point in ROC space. Considering a scoring classifier, each threshold on the scores results in a discrete classifier.

Definition 2.4. *Given a scoring classifier, the connection of* (fpr, tpr) *pairs obtained by applying all possible thresholds in decreasing order is the ROC curve.*

Figure 2.3(a) shows two ROC curves of which one is said to be dominating. For each value of fpr, the *dominating* ROC curve has an equal or higher value of tpr than the curve being dominated (Ling, Huang, and Zhang, 2003). Thus, for a fixed value of fpr, the discrete classifier constructed from the dominating ROC curve is at least as good as the one constructed from the other curve. Often, there is no dominating ROC curve. Instead, curves intersect each other and a curve is said to dominate in one or more regions of ROC space (Provost, Fawcett, and Kohavi, 1998).

An ROC curve often contains concavities, which implies local worse-than-random behaviour of the classifier (i.e., negatives are ranked before positives). Its convex hull, denoted by ROCCH, naturally removes the concavities and still connects classifiers that we can construct by thresholding scores (Fawcett, 2003). Figure 2.3(b) shows an ROC curve and its corresponding ROCCH.

Definition 2.5. *The convex hull of an ROC curve is the boundary of the minimal convex set containing all points of the ROC curve, minus the diagonal.*

Theorem 2.1. *For any point on or below the ROCCH, a classifier can be constructed by thresholding the scores in such a way that it achieves the performance statistics represented by that point.*

The proof of this theorem for the case of a point on the ROCCH is given by Provost and Fawcett (2001). Since this proof shows a way to construct specific discrete classifiers that is important for the research presented in Chapter 5, we give our own (more detailed) version of the proof below.

Proof. We fix an arbitrary point (fpr_i, tpr_i) on the ROCCH and denote the corresponding classifier as f_i. The line segment of the convex hull that contains this point is denoted by L_i. We can distinguish two cases.

- **Case 1:** (fpr_i, tpr_i) is an endpoint of L_i.
  Endpoints of line segments of the ROCCH are points that also lie on the ROC curve itself. Hence, thresholds for the classifiers corresponding to these points are found directly from the scores of the instances that are used to construct the ROC curve (a straightforward table look-up).

- **Case 2:** (fpr_i, tpr_i) is not an endpoint of L_i.
  Denote the classifiers corresponding to the endpoints of L_i by f_1 and f_2. We define $d_{1,i} = fpr_i - fpr_1$ and $d_{1,2} = fpr_2 - fpr_1$. For each instance $\boldsymbol{x}$, the classification of f_i is a random variable that takes either of values of $f_1(\boldsymbol{x})$ or

$f_2(\boldsymbol{x})$. The probabilities of these two events are $\mathbb{P}(f_i(\boldsymbol{x}) = f_1(\boldsymbol{x})) = 1 - d_{1,i}/d_{1,2}$ and $\mathbb{P}(f_i(\boldsymbol{x}) = f_2(\boldsymbol{x})) = d_{1,i}/d_{1,2}$. It follows that the expectation of *fpr* is:

$$\begin{aligned} E[fpr] &= \left(1 - \frac{d_{1,i}}{d_{1,2}}\right) fpr_1 + \frac{d_{1,i}}{d_{1,2}} fpr_2 \\ &= \frac{(fpr_2 - fpr_1)d_{1,i}}{d_{1,2}} + fpr_1 \\ &= fpr_i \ . \end{aligned}$$

Analogously, we can show that the expectation of *tpr* is equal to tpr_i. So, also in the second case, we can construct classifier f_i.

The first and second case cover all possible scenarios for points on the ROCCH, which ends the proof of the theorem. □

The case of a point below the convex hull is again a straightforward extension of interpolating classifiers on the convex hull (Flach and Wu, 2005). As a consequence, for simplicity of presentation, we will assume in the remainder of the thesis that empirical ROC curves are convex and all points can be obtained by a threshold. An ROC curve and its ROCCH are thus the same, unless stated otherwise.

The Area Under the ROC Curve

We have seen that the ROC curve is a useful tool for classifier visualisation and evaluation. However, it is often desired to have a single number to represent the performance of a scoring classifier.

For this purpose, the area under the (ROC) curve, abbreviated as AUC, is often calculated (Bradley, 1997; Ling *et al.*, 2003; Skalak, Niculescu-Mizil, and Caruana, 2007). Clearly, an optimal classifier has an AUC of 1, while a value of 0.5 is obtained by random classification (we remember that random classification is represented by the ascending diagonal in ROC space). A scoring classifier defines an ROC curve in a unique way, of course given a fixed set of instances, and therefore we will often talk about the AUC of the classifier instead of its ROC curve.

Besides the intuitive geometrical interpretation of the AUC, it received a considerable amount of attention from statistics. More specifically, the AUC of a scoring classifier is equivalent to the probability that a positive instance receives a higher score than a negative instance, both instances independently drawn from the example space $\mathcal{Z}$ according to $\mathbb{P}$. Since we cannot precisely compute this probability with only a finite set of examples, we have to settle for an estimation. Therefore, we distinguish between the *true* AUC, as defined by the unknown probability distribution, and its estimation as obtained by using the available examples, the *empirical* AUC. An unbiased estimator is the Wilcoxon-Mann-Whitney statistic, which is defined in terms of pairwise comparisons of scores (Hanley and McNeil, 1982).

Definition 2.6. *The (empirical) AUC of a scoring classifier f, given a set S with a finite number of examples, is defined as*

$$AUC(f,S) = \frac{1}{P \cdot N} \sum_{i=1}^{P} \sum_{j=1}^{N} \mathbf{1}\left[f(\boldsymbol{x}_i^p) > f(\boldsymbol{x}_j^n)\right] \quad , \tag{2.4}$$

where $\boldsymbol{x}_i^p$ and $\boldsymbol{x}_j^n$ denote the i-th positive instance and the j-th negative instance, respectively, and the term $\mathbf{1}\left[f(\boldsymbol{x}_i^p) > f(\boldsymbol{x}_j^n)\right]$ denotes the indicator function: it is 1 if the test between square brackets is true, and 0 otherwise. We remember that P and N are the number of positives and negatives in the set S, respectively.

Thus, the AUC can also be computed by counting the number of pairs of examples that are ranked correctly, i.e., pairs of which the positive instance has a higher score than the negative instance. The time complexity seems to grow quadratically with the number of instances (there are $P \cdot N$ pairs) but it has been shown that the AUC can be calculated in linear time using the rank of the instances (Hand and Till, 2001). In case there exist ties among the scores of positive and negative instances, i.e., $f(\boldsymbol{x}_i^p) = f(\boldsymbol{x}_j^n)$ for some positive and negative instance, then we have to modify (2.4) such that a value of 0.5 is added in the summation.

Examples of scoring classifiers and the corresponding AUC values will be provided in Chapter 3. Dependent on the context and what we believe to be the most illustrative, we will use either the geometrical interpretation or the statistical one.

2.3 Multi-class Classification

In this section, we first show how to solve a multi-class classification problem by means of decomposing it into a set of binary problems (2.3.1). Then, we explain how the predictions of the corresponding classifiers can be combined into a final classification of an instance by means of a simple aggregation strategy (2.3.2).

2.3.1 Learning by Pairwise Comparison

Many state-of-the-art learning algorithms can only produce classifiers that are able to distinguish between two classes. Three examples of such classifiers are: the perceptron (including its many variants), logistic regression, and the standard support vector machine. However, of course, not all classification problems are binary.

A multi-class classification problem has an unordered label space that consists of more than two elements, i.e., $m > 2$. In order to handle multi-class classification problems, it is generally accepted to decompose the problem into a series of binary classification problems for which we can learn any classifier of our choice. There exist several decomposition techniques. We focus on *learning by pairwise comparison*, which is also characterised as follows: all-pairs, one-against-one, round robin learning, and pairwise classification (Fürnkranz, 2002). The reasons for this choice are explained in the next paragraph. In learning by pairwise comparison, a separate model or *base classifier* $\mathcal{M}_{ij}$ is trained for each pair of labels $(\lambda_i, \lambda_j) \in \mathcal{L} \times \mathcal{L}$,

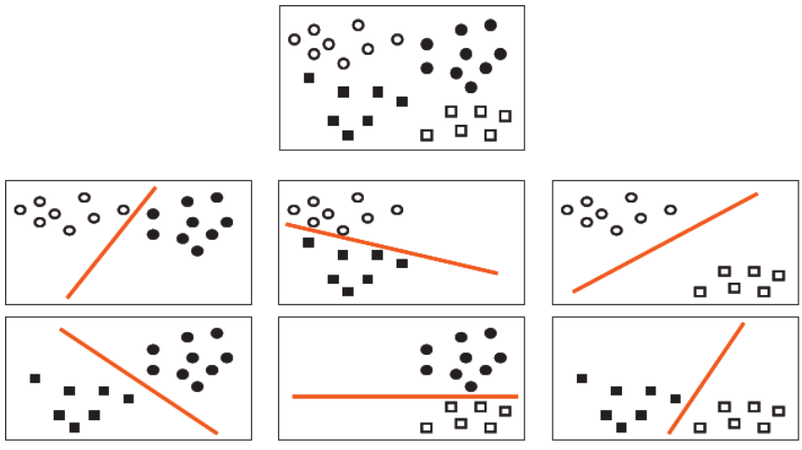


Figure 2.4: An illustration of learning by pairwise comparison. The top figure shows the instance space with four classes, each class depicted by a different symbol. The decomposition leads to six simpler and smaller binary classification problems.

$1 \leq i < j \leq m$; thus in total a number of $m(m-1)/2$ classifiers is needed. We say that the number of classifiers is quadratic in the number of classes. The base classifier $\mathcal{M}_{ij}$ is intended to discriminate instances with class label λ_i from those having class label λ_j. An illustration with a toy example is given in Fig. 2.4.

Typically, learning by pairwise comparison produces more accurate models than the alternative one-against-rest decomposition. The latter technique learns one classifier for each class using the examples of this class as positive examples and all others as negative examples. However, despite the need to train a quadratic instead of a linear number of classifiers, pairwise classification is computationally not more complex than one-against-rest. The reason is that the binary classification problems not only contain fewer training examples (because all examples that do not belong to either of the two classes are ignored), but the decision boundaries for the problems may be considerably simpler than for the problems generated by one-against-rest (Fürnkranz, 2001; 2002; Hsu and Lin, 2002). This can be seen in Fig. 2.4: if we want to separate each class from the rest, then lines cannot be used anymore (instead, we should rely on more complex models such as quadratic functions). Both techniques are special cases of a more general approach that uses error correcting output codes for the decomposition (Dietterich and Bakiri, 1995).

2.3.2 Aggregation Strategies

Given a new instance $\boldsymbol{x}$ to be classified, we receive the output of each of the base classifiers from the pairwise decomposition. So, we are in need of a strategy to aggregate the outputs $s_{ij} = \mathcal{M}_{ij}(\boldsymbol{x})$ into a single class label.

A popular *aggregation strategy* is weighted voting. This strategy considers the output s_{ij} as a weighted "vote" for class label λ_i. Correspondingly, assuming the learners to be additively reciprocal (which is natural in learning by pairwise comparison), we have that[3]

$$s_{ji} = 1 - s_{ij} \ ,$$

and this score is then considered as a weighted vote for class label λ_j. Finally, each class label λ_i is scored in terms of the sum of its votes

$$s_i = \sum_{1 \leq j \neq i \leq m} s_{ij} \ , \tag{2.5}$$

and the class label with the maximal sum of votes is predicted. Possible ties are often broken at random or are decided in favour of the majority class.

For example, consider a three-class classification problem. We learned the base classifiers $\mathcal{M}_{12}$, $\mathcal{M}_{13}$, $\mathcal{M}_{23}$ and we ask their score for the new instance. The scores are as follows (written in matrix notation):

$$[s_{ij}]_{i \neq j} = \begin{bmatrix} - & 0.90 & 0.90 \\ 0.10 & - & 0.60 \\ 0.10 & 0.40 & - \end{bmatrix} \ ,$$

where we note that the lower part of the descending diagonal is obtained by one minus the upper part (the reciprocal learners property). The vote for λ_1 is $s_1 = 0.9 + 0.9 = 1.80$, while $s_2 = 0.70$ and $s_3 = 0.50$. So, the weighted voting aggregation strategy predicts λ_1 as the class label.

2.4 Label Ranking

In this section, we review the recently introduced label ranking setting. We explain its main idea and related concepts (2.4.1) and we show that label ranking is a proper generalisation of the classification setting (2.4.2).

2.4.1 Label Rankings, Preferences, and Permutations

The setting of label ranking is among others an extension of the conventional setting of classification (Har-Peled, Roth, and Zimak, 2002; Crammer and Singer, 2003; Hüllermeier, Fürnkranz, Cheng, and Brinker, 2008). Roughly speaking, instead of associating every instance $\boldsymbol{x} \in \mathcal{X}$ with a class label, we now associate $\boldsymbol{x}$ with a complete label ranking, i.e., a total order of the class labels.

This means that we have a complete, transitive, and asymmetric relation $\succ$ on the class labels, where $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ indicates that λ_i precedes λ_j in the ranking associated with $\boldsymbol{x}$. Each class label has a unique position in the ranking and the first position is given to the class label that is considered as most important.

[3] In other work, additively reciprocal learners are sometimes called learners that satisfy pairwise consistency; see for example Ailon and Mohri (2008).

It follows that a ranking can be considered (metaphorically) as a special type of preference relation, and therefore we shall also say that $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ indicates that λ_i is *preferred* to λ_j given the instance $\boldsymbol{x}$. To illustrate, suppose that instances are descriptions of students and $\succ$ is a preference relation on a fixed set of study fields such as Math, CS, and Physics. We will use this example throughout the section.

Formally, it is convenient to identify a ranking $\succ_{\boldsymbol{x}}$ with a permutation $\tau_{\boldsymbol{x}}$ of the set $\{1, \ldots, m\}$. The class of all permutations of this set is denoted by $\mathcal{S}_m$. For ease of presentation, we define $\tau_{\boldsymbol{x}}$ in such a way that $\tau_{\boldsymbol{x}}(i) = \tau_{\boldsymbol{x}}(\lambda_i)$ is the position of class label λ_i in the ranking. This permutation encodes the ground truth ranking

$$\lambda_{\tau_{\boldsymbol{x}}^{-1}(1)} \succ_{\boldsymbol{x}} \lambda_{\tau_{\boldsymbol{x}}^{-1}(2)} \succ_{\boldsymbol{x}} \ldots \succ_{\boldsymbol{x}} \lambda_{\tau_{\boldsymbol{x}}^{-1}(m)} \quad , \tag{2.6}$$

where $\tau_{\boldsymbol{x}}^{-1}(j)$ is the index of the class label at position j in the ranking. Clearly, seen the other way around, we have that $\tau_{\boldsymbol{x}}(i) < \tau_{\boldsymbol{x}}(j)$ only if $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$. By abuse of terminology, though justified in light of the above one-to-one correspondence, we refer to elements $\tau \in \mathcal{S}_m$ as both permutations and rankings.

2.4.2 Generalisation of Classification

In analogy to the conventional classification setting, we do not assume that there exists a deterministic mapping from instances to permutations. Instead, every instance is associated with a *probability distribution* over $\mathcal{S}_m$. This means that there exists a probability distribution $\mathbb{P}(\cdot \,|\, \boldsymbol{x})$ for each $\boldsymbol{x} \in \mathcal{X}$ such that, for every $\tau \in \mathcal{S}_m$,

$$\mathbb{P}(\tau \,|\, \boldsymbol{x}) \tag{2.7}$$

is the probability that $\tau_{\boldsymbol{x}} = \tau$ (each permutation has a probability that it is the correct permutation for the instance). As an illustration, going back to our example, the following probability distribution may be given for a particular student:

label ranking τ					$\mathbb{P}(\tau \,\|\, \boldsymbol{x})$
Math	$\succ$	CS	$\succ$	Physics	.4
Math	$\succ$	Physics	$\succ$	CS	.3
CS	$\succ$	Math	$\succ$	Physics	.0
CS	$\succ$	Physics	$\succ$	Math	.2
Physics	$\succ$	Math	$\succ$	CS	.0
Physics	$\succ$	CS	$\succ$	Math	.1

Such a probability distribution can be used to reduce a ranking to a single class label prediction. To see this, we remember that in the setting of conventional classification, training data consists of tuples $(\boldsymbol{x}_k, \lambda_{\boldsymbol{x}_k})$ which are assumed to be produced according to a probability distribution over $\mathcal{Z} = \mathcal{X} \times \mathcal{L}$. This implies that we can associate with each instance $\boldsymbol{x}$ a vector of conditional probabilities

$$p_{\boldsymbol{x}} \;=\; (\,\mathbb{P}(\lambda_1 \,|\, \boldsymbol{x}), \ldots, \mathbb{P}(\lambda_m \,|\, \boldsymbol{x})\,) \quad , \tag{2.8}$$

where $\mathbb{P}(\lambda_i \,|\, \boldsymbol{x})$ denotes the probability of observing the class label $\lambda_{\boldsymbol{x}} = \lambda_i$ given the instance. Now, in label ranking, the class label $\lambda_{\boldsymbol{x}}$ can be naturally associated

with the top label in the ranking $\tau_{\boldsymbol{x}}$, i.e., $\lambda_{\boldsymbol{x}} = \tau_{\boldsymbol{x}}^{-1}(1)$. In other words, $\mathbb{P}(\lambda_i \,|\, \boldsymbol{x})$ corresponds to the probability that λ_i occurs as a top label in a ranking τ. It is computed by summing the probabilities of all possible rankings in which the class label is at the first (top) position. In our example, this yields probabilities:

$$\mathbb{P}(\text{Math} \,|\, \boldsymbol{x}) = .7, \;\; \mathbb{P}(\text{CS} \,|\, \boldsymbol{x}) = .2, \;\; \mathbb{P}(\text{Physics} \,|\, \boldsymbol{x}) = .1 \;\; .$$

To show that the label ranking setting is a proper generalisation of the conventional setting of classification, it is also necessary to have a mapping in the reverse direction, i.e., a mapping from probability vectors (2.8) to measures (2.7). Such a mapping can be defined in different ways. Since the order of the non-top labels is irrelevant in the classification setting, it appears reasonable to distribute the probability mass $\mathbb{P}(\lambda_i \,|\, \boldsymbol{x})$ equally on $\{\tau \in \mathcal{S}_m \,|\, \tau^{-1}(1) = \lambda_i\}$. The result is an inverse mapping expressing "indifference" with respect to the order of non-top labels.[4] In our example, this gives the following probability distribution over the rankings:

label ranking τ					$\mathbb{P}(\tau \,|\, \boldsymbol{x})$
Math	$\succ$	CS	$\succ$	Physics	.35
Math	$\succ$	Physics	$\succ$	CS	.35
CS	$\succ$	Math	$\succ$	Physics	.10
CS	$\succ$	Physics	$\succ$	Math	.10
Physics	$\succ$	Math	$\succ$	CS	.05
Physics	$\succ$	CS	$\succ$	Math	.05

More details on the label ranking setting will be given in Chapter 6. There, we will also conjecture that analysing learning by pairwise comparison is more natural in the label ranking setting than it is in the conventional classification setting.

2.5 Chapter Summary

This chapter provided general background information in support of the research that we present in this thesis. We focussed on supervised machine learning.

Dependent on the characteristics of the label space, we have distinguished between the classification learning setting and the label ranking setting. The classification learning setting was discussed for binary and multi-class problems. Concerning the binary problems, we reviewed discrete classifiers and scoring classifiers (yielding a bipartite ranking). The ROC curve and the area under the curve have been discussed as important tools for classifier performance evaluation. Concerning the multi-class problems, we explained learning by pairwise comparison and aggregation strategies. Finally, at the very end of this chapter, we introduced the label ranking setting which recently received a great deal of attention in machine learning.

In the next chapters, many of these concepts will return. For ease of readability and not to overwhelm the reader with notation, each of the following chapters contains a short recapitulation of some of the above concepts and settings.

[4] The concept of indifference is interesting from a reliability point of view. For more information, we refer to Hüllermeier and Brinker (2008) and Hühn and Hüllermeier (2009).

Chapter 3

The Effectiveness of the AUC

The AUC has been widely used to assess the ranking performance of binary scoring classifiers. From a model evaluation and selection point of view, several authors have recently argued for enhancements of the AUC that take the difference between scores into account. Small differences should be treated more carefully. In this chapter, we investigate whether these variants of the AUC are indeed more effective performance metrics when they are used in a model selection scenario. Our result provides an answer to the first research question.[1]

3.1 Selecting the Best Ranker

Traditionally, machine learning classifiers are used in the common supervised classification setting where the goal is to find an accurate mapping from instance space to label space. However, in many applications, it is not sufficient to predict the most likely class label for each test instance (Cohen, Shapire, and Singer, 1997). What is needed instead is, assuming a binary label space for simplicity, a total ordering of test instances from most likely positive to most likely negative. In this way a *ranking* of test instances is obtained. For example, consider a system that identifies the top ten percent of recently convicted offenders that are most likely to commit the same or similar crime in a short time period after the conviction. Or, as a second example, consider a credit-card fraud detection system that ranks bank accounts according to the possibility of being severely compromised. For more details and illustrations we refer to the relevant parts in Chapters 1 and 2.

The rank position of an instance depends on the score that it receives from the (scoring) classifier. The area under the ROC curve, or simply AUC, has been widely used to assess the ranking performance of a classifier, i.e., to measure how good it ranks instances. We note that the AUC can also be calculated without an explicit construction of the ROC curve. As an illustration, we consider the following two

[1] This chapter is based on an article by Vanderlooy and Hüllermeier (2008) which was selected among the 7 out of 521 ECML/PKDD 2008 submissions for publication in the journal special issue.

classifiers, f_1 and f_2, and their scores on a *validation set* consisting of four positive ($\boldsymbol{x}_1$ to $\boldsymbol{x}_4$) and three negative instances ($\boldsymbol{x}_5$ to $\boldsymbol{x}_7$):

	$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$	$\boldsymbol{x}_6$	$\boldsymbol{x}_7$
	|	|	|	|	|	|	|
f_1 :	1.0+	1.0+	0.52+	0.4+	0.5−	0.0−	0.0−
f_2 :	1.0+	1.0+	0.82+	0.4+	0.5−	0.0−	0.0−

Both classifiers give the maximum score of 1 to two positive instances ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$) and the minimum score of 0 to two negative instances ($\boldsymbol{x}_6$ and $\boldsymbol{x}_7$). Also, there is a negative instance ($\boldsymbol{x}_5$) with a higher score than one positive instance ($\boldsymbol{x}_4$). We note that, in total, there are twelve pairs consisting of a positive instance and a negative instance. Only one of these pairs is *incorrect* in the sense that the negative instance has higher score than the positive instance. Therefore, we have that both classifiers have an AUC value of $11/12 \approx 0.92$. Hence, the AUC is defined such that it only considers the sign of the differences between scores of pairs of positive and negative instances, while it ignores the absolute value of the score differences.

Consequently, it can happen that a small change in scores leads to a considerable change in AUC value. Such an effect is especially apparent when the number of instances used to calculate the AUC is small and/or the class distributions are overlapping to some degree. Going back to our illustration above, we consider that there has occurred a small change in classifiers f_1 and f_2. For f_1, the result is that the scores of 0.52 and 0.5 are interchanged among the corresponding instances, meaning that $f_1(\boldsymbol{x}_3) = 0.5$ and $f_1(\boldsymbol{x}_5) = 0.52$. The scores of the other instances are of course also altered but the pairwise orderings remain as they are. This is reasonable since corresponding score differences are large, meaning that their sign is not easily altered. So, looking at the net effect, we have that one previously correctly ordered instance pair has become incorrectly ordered. Therefore, the AUC value of the classifier becomes $10/12 \approx 0.83$ (a decrease in value of almost 10%). In contrast, considering f_2, small changes in the scores will always lead to the same AUC value since the larger score differences ensure that pairwise orderings remain as they are.

In a similar fashion we can easily construct examples where a small change in scores leads to a large improvement in AUC value. We can also imagine two classifiers with the same AUC, even though it may be intuitively reasonable to select one of them as a "better separator" in the sense that the classifier increases the difference between scores of positive and negative instances, respectively. Hereby, it seems to be more certain in its decisions and it is expected to perform alike among various test sets. We note that, since we are interested in selecting the best classifier (called model) among a set of candidates, we are in a *model selection* scenario.

With the examples just mentioned, it has been argued that the insensitivity of the AUC toward score differences is disadvantageous for model evaluation and selection. For this reason, three variants of the AUC metric that take the score differences into account have recently been proposed, along with first experimental results (Ferri, Flach, Hernández-Orallo, and Senad, 2005; Wu, Flach, and Ferri, 2007; Calders and Jaroszewicz, 2007). The main setup of these experiments is a model selection scenario where the AUC and its variants are used to select a single model out of

a set of candidate models. The goal is to select the model with highest AUC as measured on an independent test set. Clearly, the metric that most often chooses the best ranker is preferred. We will follow this setup of proper model selection.

In this chapter we are interested in thoroughly comparing the conventional AUC and its three variants. Hence, we are interested in answering our first research question (RQ 1): *To what extent is the area under the ROC curve an effective performance metric for bipartite ranking when compared to its variants that consider the absolute values of the scores?* To answer this research question, we first present a formal analysis that leads us to conjecture that actually none of the variants should be able to outperform the conventional AUC (with regard to model selection). This conjecture is then verified empirically on the basis of experiments with synthetic data and real benchmark data. Even though we do not invalidate previous experiments, our empirical study is arguably more extensive, especially since it considers different types of model selection scenarios. The superior effectiveness of the AUC as a ranking performance metric becomes clear dependent on the scenario. Finally, our contribution also sheds light on recent research dealing with the construction of classifiers that try to (approximately) optimise the AUC.

The remainder of the chapter is organised as follows. In Section 3.2 we consider the AUC from a statistical viewpoint. Then, in Section 3.3, we present the unified framework and explain the three variants of the AUC metric. In Section 3.4 we analyse these metrics in a formal way. An extensive experimental verification of our conjecture is provided in Section 3.5. Implications of our contribution to classifier learning are given in Section 3.6 and we discuss related work in Section 3.7. Finally, Section 3.8 provides a conclusion and an answer to the research question.

3.2 AUC as an Estimator for Ranking Performance

Using the terminology introduced in Chapter 2, we consider an instance space $\mathcal{X}$ and let the example space $\mathcal{Z} = \mathcal{X} \times \{+1, -1\}$ be endowed with an unknown probability measure $\mathbb{P}$. Thus, $\mathbb{P}(\boldsymbol{x}, \lambda_{\boldsymbol{x}})$ denotes the probability to observe instance $\boldsymbol{x}$ with class label $\lambda_{\boldsymbol{x}}$. An instance with class label $+1$ (-1) is called a positive (negative) instance. We refer to a scoring classifier f as an $\mathcal{X} \to [0, 1]$ mapping and $f(\boldsymbol{x})$ is interpreted as the probability or, more generally, as a degree of confidence that the class label of $\boldsymbol{x}$ is $+1$. Often, we will use the term model instead of classifier in order to be consistent with the large literature about model selection.

A natural performance metric for ranking is the probability of the event $f(\boldsymbol{a}) > f(\boldsymbol{b})$ given that $\boldsymbol{a}$ is a positive instance and $\boldsymbol{b}$ is a negative instance, both randomly drawn from the example space. Empirically, this probability has to be estimated from a sample $S = \{(\boldsymbol{x}_i, \lambda_{\boldsymbol{x}_i})\}_{i=1}^{s} \subseteq (\mathcal{X} \times \{+1, -1\})^s$. An unbiased estimator is the Wilcoxon-Mann-Whitney statistic, which is given by the fraction of pairs $(\boldsymbol{x}_i^p, \boldsymbol{x}_j^n)$ with $\boldsymbol{x}_i^p$ a positive instance and $\boldsymbol{x}_j^n$ a negative instance for which $f(\boldsymbol{x}_i^p) > f(\boldsymbol{x}_j^n)$ (Mann and Whitney, 1947). So, we simply count the number of pairs of instances that are correctly ordered. In case of a tie in the scores, $f(\boldsymbol{x}_i^p) = f(\boldsymbol{x}_j^n)$, the instance pair is counted as $1/2$ instead of as 1.

Interestingly, the above statistic is equivalent to the AUC (Hanley and McNeil, 1982; Bradley, 1997). Obviously, an optimal classifier (i.e., a perfect ranker) has an AUC value of 1 while a value of 0.5 is obtained for a random classifier. An advantage of the AUC over many other performance metrics is that it is invariant to changes in cost and class distributions. Also, even though the computational cost seems to grow quadratically in the number of instances, the metric can be computed in time complexity $\mathcal{O}(|S| \log |S|)$ by sorting the instances and keeping track of the ranks of the positives (Hand and Till, 2001).

3.3 AUC and its Variants

In this section, we introduce a generalisation of the AUC that allows us to analyse the AUC and its variants in a unified way (3.3.1). The three variants are denoted by scored AUC (3.3.2), soft AUC (3.3.3), and probabilistic AUC (3.3.4).

3.3.1 Generalisation of the AUC

Given a sample S, consider the following generalisation of the estimate of the AUC:

$$\text{gAUC}(f, S) = \frac{1}{P \cdot N} \sum_{i=1}^{P} \sum_{j=1}^{N} w\left(f(\boldsymbol{x}_i^p) - f(\boldsymbol{x}_j^n)\right) \quad , \tag{3.1}$$

where $\boldsymbol{x}_i^p$ and $\boldsymbol{x}_j^n$ denote the i-th positive instance and the j-th negative instance, respectively (no specific ordering of the instances is assumed). We call the function $w(\cdot)$ the *modifier function*. It is an arbitrary $[-1, 1] \to [0, 1]$ mapping that defines how to account for differences between the scores of positive instances and negative instances. For the conventional AUC metric, $w(\cdot)$ is given by (see Fig. 3.1(a))

$$w(t) = w^*(t) = \begin{cases} 1 & \text{if } t > 0 \\ 0.5 & \text{if } t = 0 \\ 0 & \text{if } t < 0 \end{cases} \quad . \tag{3.2}$$

Henceforth, we refer to pairs $(\boldsymbol{x}_i^p, \boldsymbol{x}_j^n)$ as PN-pairs and the differences in scores $f(\boldsymbol{x}_i^p) - f(\boldsymbol{x}_j^n)$ are denoted *score margins*. A PN-pair is said to be correctly ordered or concordant when $f(\boldsymbol{x}_i^p) > f(\boldsymbol{x}_j^n)$; when the score margin is negative, it is discordant. Hence, with this terminology, we can say that the estimate of the AUC is calculated by counting the number of concordant PN-pairs and the number of PN-pairs with zero score margin. The PN pairs in the first group contribute each a count of 1 while the PN-pairs in the second group contribute each a count of $1/2$.

3.3.2 Variant 1: scored AUC

The value of the AUC is invariant with respect to the score margin as long as its sign remains unchanged. However, as mentioned in Section 3.1, it appears intuitively reasonable to prefer larger score margins to smaller ones. Moreover, it was argued

by Wu *et al.* (2007) that the concordance of a PN-pair is in a sense less reliable when the score margin is small. In order to make the evaluation of a classifier more robust, the authors proposed a variant of the AUC metric (called scored AUC) that takes the absolute value of the score margins into account.

The key idea is to count a PN-pair only when the score margin exceeds a threshold $\tau \in [0, 1]$. It follows that the conventional AUC is recovered for $\tau = 0$, while the modified AUC is a decreasing function of τ. A value of 0 is obtained for $\tau = 1$.[2] The effect of different thresholds can be visualised by plotting each τ against the modified AUC. As a result, a piecewise linear curve is obtained in which the drops occur at the τ's that equal the score margin of one of the PN-pairs. The area under this curve aggregates the robustness (or sensitivity) of the conventional AUC over all possible thresholds, hereby preventing the user from committing to a single τ. The area is called the *scored* AUC (sAUC). It is not difficult to show that the sAUC equals (3.1) with the following modifier function (see Fig. 3.1(b)):

$$w(t) = \begin{cases} t & \text{if } t > 0 \\ 0 & \text{if } t \leq 0 \end{cases} \quad . \tag{3.3}$$

Thus, the contribution of a PN-pair $(\boldsymbol{x}_i^p, \boldsymbol{x}_j^n)$ to the evaluation of a classifier is the score margin $t = f(\boldsymbol{x}_i^p) - f(\boldsymbol{x}_j^n)$ if this score margin is positive, and 0 otherwise. A simple decomposition of the metric shows that it can be computed in linear time.

3.3.3 Variant 2: soft AUC

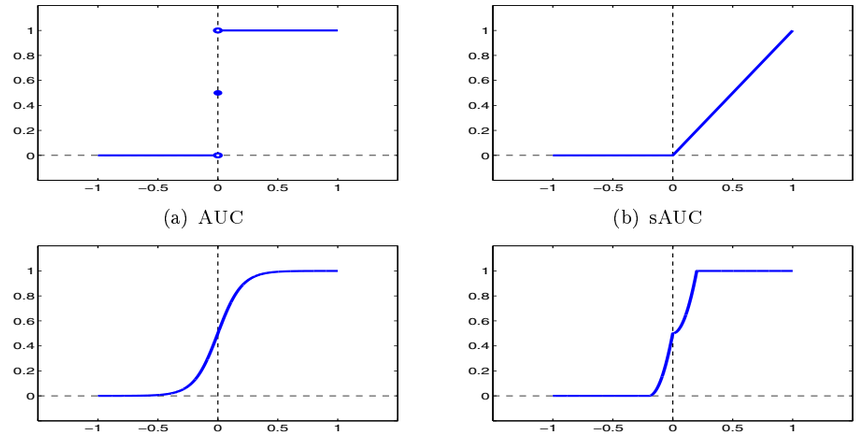
We have seen that the scored AUC punishes classifiers that produce small score margins, as these are considered as uncertain, and therefore may contribute to the AUC just by chance. A second variant of the AUC metric, called softAUC, has originally been proposed as a differentiable approximation of AUC amenable to learning algorithms requiring a continuous objective function. For example, it can be used by a gradient descent routine to find a hyperplane that approximately maximises the AUC (Calders and Jaroszewicz, 2007). Nonetheless, the softAUC also fits the purpose of this chapter and it can be represented as a special case of (3.1) using a sigmoidal modifier function (see Fig. 3.1(c)):

$$w(t) = \frac{1}{1 + \exp(-\beta t)} \quad , \tag{3.4}$$

with $\beta \in \,]\,0, \infty\,[$. We note that a large β implies that the sigmoid approximates the step function, i.e., softAUC converges to conventional AUC for $\beta \to \infty$.

The sigmoid function automatically smoothes out the region around the zero score margin. Its computational cost is quadratic, but accurate approximations that are computable in linear time can be used as an alternative (Herschtal and Raskutti, 2004; Calders and Jaroszewicz, 2007).

[2] We note that the effect of a particular τ can also be seen as smoothing the ROC curve along the parts where the score margins are not large enough. Although this clearly is an interesting interpretation, we do not elaborate further on this viewpoint.



(a) AUC (b) sAUC

(c) softAUC (d) probAUC

Figure 3.1: Modifier functions that are used by the different AUC metrics.

3.3.4 Variant 3: probabilistic AUC

A third and last variant is the probabilistic AUC (probAUC). The key idea of this metric has been introduced by Ferri *et al.* (2005), but it has not been elaborated further at the present time. Yet, it is a more rigorous realisation of the idea also underlying sAUC, namely that a score generated by a classifier is considered as a "noisy" observation of the true score of the instance.

Given an estimated score $f(\boldsymbol{x}_i)$, the true score of the instance is modelled as a random variable uniformly distributed in $[f(\boldsymbol{x}_i) - h, f(\boldsymbol{x}_i) + h]$. Then, given a positive instance with a score in $[a - h, a + h]$ and a negative instance with a score in $[b - h, b + h]$, the probability that this PN-pair is concordant is[3]

$$\int_{a-h}^{a+h} \int_{b-h}^{x} (2h)^{-2} \, dy \, dx \ ,$$

where the two instances are drawn according to the conditional distributions $\mathbb{P}(\cdot \mid \lambda_{\boldsymbol{x}} = +1)$ and $\mathbb{P}(\cdot \mid \lambda_{\boldsymbol{x}} = -1)$, respectively. Clearly, the above probability only depends on $t = a - b$ and is given by (see Fig. 3.1(d))

$$w(t) = \begin{cases} 1 & \text{if } t \geq h \\ \max\left(0, \frac{t}{2h}\right) + \frac{1}{2}\left(1 - \frac{|t|}{2h}\right)^2 & \text{if } -h < t < h \\ 0 & \text{if } t \leq -h \end{cases} \ . \tag{3.5}$$

We denote by probAUC the generalisation (3.1) with (3.5) as a modifier function.

[3]Here, we ignore that the boundary cases $a > 1 - h$ and $a < h$ (and the same cases for b) do actually need special treatment.

It is clear that the width $2h$ of the interval defines the level of smoothing, and probAUC converges to the conventional AUC for $h \to 0$. Instead of assuming a uniform distribution, other distributions such as a truncated Gaussian or a triangular can of course be considered. The computational cost using a uniform distribution grows quadratically in the number of instances.

3.4 Formal Analysis of the AUC Metrics

In this section, we present our formal analysis of the four aforementioned AUC metrics. We start by identifying some potential problems and critical issues concerning the three variants (3.4.1). Then, we present a bias-variance analysis, leading to our conjecture that none of the variants is as effective as the AUC itself (3.4.2). Experiments with synthetic data illuminate and verify this conjecture (3.4.3). We end this section with some conclusions from the formal results (3.4.4).

3.4.1 Potential Problems and Critical Issues

From the four modifier functions shown in Fig. 3.1, it is obvious that sAUC is the most extreme modification of the AUC. However, it is also the variant that has been investigated, with regard to model evaluation and selection, most thoroughly by means of experimental studies (Wu *et al.*, 2007).

Independent of this, we would like to point out two potential disadvantages of sAUC. The first one is a kind of asymmetry of its modifier function (3.3) which considers concordant PN-pairs with small score margin as potentially discordant pairs, but not the other way round. That is, PN-pairs of which the score margin is negative and small in absolute value (the negative instance in a pair has a slightly higher score than the positive instance) are not considered as potentially concordant pairs. The second problem is that, by aggregating over all possible margins, sAUC implicitly assumes that the classifiers to be evaluated produce scores in the same range. As a consequence, a classifier that produces scores close to the extreme values 0 and 1 is likely to be preferred over a classifier that produces less extreme values, even if the latter makes fewer ranking mistakes. As an illustration, consider the following two classifiers and their scores on a validation set consisting of four positive ($\boldsymbol{x}_1$ to $\boldsymbol{x}_4$) and three negative instances ($\boldsymbol{x}_5$ to $\boldsymbol{x}_7$):

	$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{x}_4$	$\boldsymbol{x}_5$	$\boldsymbol{x}_6$	$\boldsymbol{x}_7$
	|	|	|	|	|	|	|
f_1 :	0.7+	0.7+	0.7+	0.7+	0.3−	0.3−	0.3−
f_2 :	1.0+	1.0+	1.0+	0.0+	1.0−	0.0−	0.0−

The classifier f_1 ranks all positive instances before all negative instances, i.e., it is a perfect ranker and therefore has maximal AUC. Yet, it has a low sAUC value (0.4 to be precise). Classifier f_2 has lower AUC since it gives a score of 0 and 1 to a positive

($\boldsymbol{x}_4$) and negative instance ($\boldsymbol{x}_5$), respectively. Its sAUC value is however 0.5.[4] Thus, in this example, model selection based on sAUC would clearly lead to a questionable choice. In fact, assuming that the sample is representative of the population, it leads to an incorrect choice. We note that the other two variants, softAUC and probAUC, overcome the first problem and strongly alleviate the second problem.

Finally, we remark that, in our opinion, the idea of considering a score $f(\boldsymbol{x}_i)$ as a kind of uncertainty measurement (random variable) lacks a convincing theoretical justification. In fact, when the classifier is fixed, as is the case in model evaluation, then the scores produced for instances $\boldsymbol{x}_i$ are determined in a deterministic way. Of course, changing the instances, i.e., evaluating a classifier on a different validation set, will also change the scores. Here, however, the randomness is introduced by the selection of the $\boldsymbol{x}_i$ and not by their scoring. Consequently, statistical properties of sampling are transferred to properties of scoring. In particular, assuming that the validation set is a representative sample, the obtained set of scores $f(\boldsymbol{x}_i)$ is also a representative sample of all scores produced by f. This point immediately leads us to the next subsection where we present a bias and variance analysis of the metrics.

3.4.2 Bias and Variance of Estimation

Recall that the goal in model selection is to select, among a set of candidates, a single model of which the *true* AUC is highest. From a statistical viewpoint, the empirical AUC of a model f on a validation set S is clearly a good estimator, especially since it is an unbiased estimate of the true AUC value.

As opposed to this, it is obvious that $\mathrm{sAUC}(f, S) \leq \mathrm{AUC}(f, S)$ for all classifiers f, and often the inequality will be strict. Therefore, sAUC produces a biased estimate. More interestingly, since less obviously, even the symmetric modifier function used by softAUC will usually produce a biased estimate. To see this, we note that the expected value of a modified AUC metric is given by the expected value of $w(T)$, where T is the score margin for a randomly chosen PN-pair. Denote the cumulative distribution function (CDF) of T by $G(\cdot)$, i.e., $G(t) = \mathbb{P}(T \leq t)$, and let $g(\cdot)$ be the corresponding probability distribution function (PDF). The expected value is then

$$\mathbb{E}(w(T)) \;=\; \int_{-1}^{1} w(t)\, dG(t) \;=\; \int_{-1}^{1} w(t) g(t)\, dt \;\; .$$

Now, for any reasonable classifier f, one may expect that the PDF $g(\cdot)$ is monotone increasing, which means that higher score margins are not less probable than lower margins. We assume that $g(t) \geq g(-t)$ for all $t \geq 0$, which is an even weaker assumption that allows for concavities in the PDF; it may be natural to expect locally this behaviour (Fawcett and Niculescu-Mizil, 2007). Using the property $w(-t) \leq 1 - w(t)$, which can easily be shown to hold for all three AUC variants, the difference

[4] In fact, for f_2, the $\tau \mapsto \mathrm{AUC}(\tau)$ function is simply a horizontal line at the height of the conventional AUC value.

between $\mathbb{E}(w(T))$ and the expected value of the conventional AUC can be bounded:

$$\begin{aligned}
\mathbb{E}(w(T)) - \mathbb{E}(w^*(T)) &= \int_{-1}^{1} (w(t) - w^*(t))g(t)\,dt \\
&= \int_{0}^{1} w(-t)g(-t) + (w(t) - 1)g(t)\,dt \\
&\leq \int_{0}^{1} (1 - w(t))g(-t) - (1 - w(t))g(t)\,dt \\
&= \int_{0}^{1} \underbrace{(1 - w(t))}_{\geq 0}\underbrace{(g(-t) - g(t))}_{\leq 0}\,dt \\
&\leq 0 \ .
\end{aligned}$$

Thus, the true AUC is underestimated by all three variants of the AUC.

Of course, biased estimation is not disadvantageous per se. First, with regard to model selection, a bias does actually not have any influence as long as it is constant, i.e., independent of the model. However, this is not guaranteed in our context since the PDF $g(\cdot)$ depends on the classifier f, and therefore is model-specific. Second, it is known in statistics that, by biasing an estimation, it is sometimes possible to reduce variance and, thereby, to obtain more precise estimations (Friedman, 1997). Indeed, since $w(-t) \leq 1 - w(t)$ in conjunction with $0 \leq w(t) \leq 1$ also implies $w(-t)^2 \leq 1 - w(t)^2$, we can show in the same way as above that

$$\mathbb{E}(w(T)^2) - \mathbb{E}(w^*(T)^2) \leq 0 \ ,$$

and consequently comparing the variances gives:

$$\mathbb{V}(w(T)) - \mathbb{V}(w^*(T)) = \underbrace{(\mathbb{E}(w(T)^2) - \mathbb{E}(w^*(T)^2))}_{\leq 0} - \underbrace{(\mathbb{E}^2(w(T)) - \mathbb{E}^2(w^*(T)))}_{\leq 0} \ .$$

This means that the change in variance can go into both directions. Examples for these two cases in terms of suitable PDFs $g(\cdot)$ can easily be constructed for any modifier function $w(\cdot)$. As an illustration, consider the softAUC modifier function (3.4) and a perfect classifier, i.e., the PDF assigns zero probability to negative score margins and is monotone increasing in positive score margins. Then the variance of the AUC is zero since it will output always a value of 1 while the softAUC in fact has a variance. To illustrate the other case, consider that $g(\cdot)$ can only take two values at $-\varepsilon$ and $+\varepsilon$ with $0 < \varepsilon \ll 1$. Let us say that the corresponding probabilities are 0.45 and 0.55. Then, clearly, the variance of the AUC is larger than that of softAUC and the difference increases with decreasing β in (3.4).

In summary, we may conclude that all three AUC variants produce estimates of the true AUC with a non-constant bias, and this is a disadvantage in model selection. Nonetheless, they may potentially reduce variance, which would of course be an advantage. First, however, this is not guaranteed since they may as well increase variance. Second, this potential advantage will only be relevant for relatively small

data sets, as the variance of the estimate decreases with sample size and, hence, becomes less important. All things considered, we do not see strong reasons to believe that any of the variants should outperform the conventional AUC in model evaluation and selection. On the contrary, we conjecture that the conventional AUC will show superior performance in this regard. We shall corroborate this conjecture by means of several experimental studies in the next parts of this chapter.

3.4.3 Simulation Studies with Synthetic Data

In this subsection, we describe the experimental setup and results from two simulation studies with synthetic data. For these studies, we have parametrised the softAUC with $\beta = 3$ and $\beta = 10$ in (3.4) to investigate the parameter influence. The results for probAUC are omitted for ease of presentation since probAUC behaves very similar to softAUC (indeed, as can be seen in Fig. 3.1, the corresponding modifier functions have strong resemblance).

In a first simulation experiment, we have simulated a classifier producing scores according to an exponential PDF truncated to $[0, 1]$:

$$\mathbb{P}(f(\boldsymbol{x}_i) \,|\, \lambda_{\boldsymbol{x}_i}) = \frac{\alpha}{1 - \exp(-\alpha)} \cdot \exp\left(-\alpha \; d(\boldsymbol{x}_i)\right) \quad ,$$

where $d(\boldsymbol{x}_i) = 1 - f(\boldsymbol{x}_i)$ if $\lambda_{\boldsymbol{x}_i} = +1$ and $d(\boldsymbol{x}_i) = f(\boldsymbol{x}_i)$ if $\lambda_{\boldsymbol{x}_i} = -1$. The value of α determines the "strength" of the associated classifier: higher values decrease the probability of an incorrectly ordered PN-pair. In total, we generated 50 scores for positive instances and 50 scores for negative instances, and afterward computed the four AUC metrics on this sample. The experiment was repeated 5000 times to approximate expected values by averages. In Fig. 3.2, we show the bias and variance of the obtained values of the AUC metrics as a function of the parameter $\alpha = 1, 2, \ldots, 10$. In agreement with our theoretical results, we find that the AUC variants are indeed underestimates of the conventional AUC. As expected, the bias reduces when the strength of the classifier is increased since the generated scores lie close to the boundaries 0 and 1. It is also clear that, for sAUC, not only the bias but also the variance remains high, even for large α. We also see that a larger β for softAUC gives better results, verifying our conjecture that the AUC is still the best performance metric (recall that softAUC converges to the AUC for $\beta \to \infty$). A similar claim can be made for probAUC since it converges to the AUC for $h \to 0$.

In a second simulation experiment, we mimic a model selection scenario as follows. A data set is randomly generated with each of its two features in the interval $[-1/\sqrt{2}, +1/\sqrt{2}]$. In this way, the perpendicular distance of a test instance to the linear model lies in the interval $[0, 1]$ and therefore can be used as a score without further modification. A linear model defined by a random weight vector and passing through the origin, $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}$, is used to label these instances. Two other suboptimal models are included in the model selection by adding Gaussian noise to the weight vector. Moreover, we randomly switch 10% of the labels of the positives and the negatives to make the selection harder. Afterward, we compute the AUC metrics for each of the three models. This experiment is repeated for 1000 times and

the number of instances ranges from 40 to 2000. In Fig. 3.3 we show the average number of times that each AUC metric selects the best model. It is clear that sAUC performs extremely poor throughout the complete setup, while the other variants can be considered as competitive to the conventional AUC.

3.4.4 Conclusions from the Theoretical Results

From our theoretical analysis we may conclude that the AUC variants are all biased and their variance can go in either direction. The net effect on the quality of the estimations is thus not clear and, hereby, there is no solid theoretical foundation for the variants. In addition, we have conjectured that the conventional AUC metric is still the most effective performance metric for model evaluation and selection. Experiments with synthetic data confirm this conjecture. In the next section we provide even stronger evidence for our conjecture by means of an extensive set of experiments on real benchmark data.

3.5 Experimental Analysis of the AUC Metrics

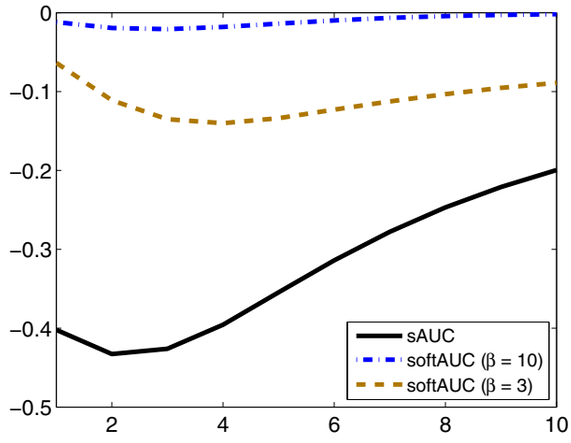
In this section, we present two sets of experiments on benchmark data sets to corroborate our formal analysis. We explain the experimental setup (3.5.1) and describe the results of the two experiments (3.5.2 and 3.5.3).

3.5.1 Experimental Setup

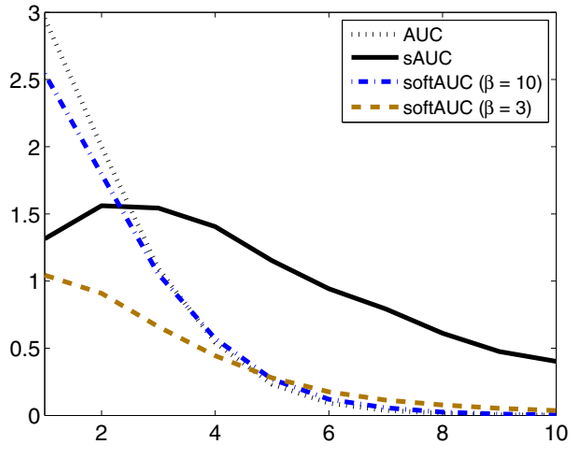
We used 16 binary benchmark data sets from the UCI repository (Asuncion and Newman, 2007). These data sets and their most important characteristics are given in Table 3.1. We note that the data sets contain at most 1000 instances since, as already mentioned earlier, a positive effect of the modified versions of the AUC, if any, is to be expected only for small data sets. For all experiments we used the `WEKA` software (Witten and Frank, 2005).

We assess the performance of AUC, sAUC, softAUC, and probAUC as model selection criteria for two different settings. In the first setting, we replicated the experimental setup of the paper by Wu *et al.* (2007) where the sAUC was introduced. Using this setup, the authors were able to show that this variant can indeed outperform the AUC, albeit with very small differences. Our results are largely in agreement with these experiments. However, by analysing the results in more detail, we may also conclude that they must be considered with reservation due to the special characteristics of the setting. Therefore, we performed a second experimental study using a setting with different and arguably more realistic characteristics.

For both settings, the parameters for softAUC and probAUC are $\beta = 10$ and $h = 0.1$, respectively. We determined these parameters such that the modifier functions behave steeply around the region of zero score margin, although not too steep to remain distinguishable from the conventional AUC.



(a)



(b)

Figure 3.2: Results of each AUC metric obtained in the first simulation experiment: (a) the bias, and (b) the variance with scale y-axis equal to 10^{-3}. We especially note that the bias and variance of sAUC remains high throughout the complete setup. Also, for softAUC, a steep modifier function gives the best results.

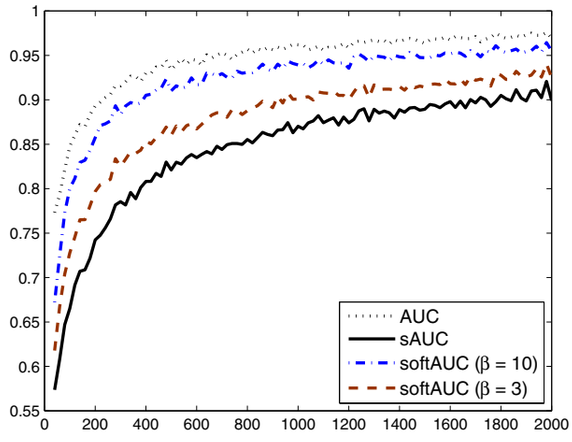


Figure 3.3: Average fraction that each AUC metric selects the best model in the second simulation experiment. The sAUC performs worst throughout the setup.

Table 3.1: The sixteen data sets, where the column headings refer to: (1) reference number, (2) data set name, (3) number of instances, (4) number of nominal features, (5) number of numerical features, and (6) percentage of the majority class.

#	name	size	nom	num	% maj class
1	breast cancer	286	9	0	70.28
2	credit rating	690	9	6	55.51
3	german credit	1000	13	7	69.40
4	heart statlog	270	7	6	59.50
5	horse colic	368	14	7	63.04
6	house votes	435	16	0	38.62
7	ionosphere	351	0	34	35.90
8	liver	345	1	5	42.03
9	monks1	556	6	0	50.00
10	monks2	604	6	0	65.72
11	monks3	554	6	0	55.41
12	pima	768	0	8	65.10
13	sonar	208	0	60	53.36
14	spect	267	22	0	58.80
15	tic-tac-toe	958	9	0	65.34
16	wisconsin breast	699	0	8	65.52

3.5.2 Setting 1: Changing the Training Data Sets

A data set is partitioned into two equal-sized parts using a stratified split.[5] One half is used as a training set and the other half is partitioned (again stratified) into 20% validation set and 80% test set. Ten different classifiers are trained with the same learning algorithm by randomly removing three features before training. The best classifier is then selected according to each of the four AUC metrics using the validation set. Finally, the performance of each selected model is assessed by comparing its AUC on the test set with that of the true best classifier. Henceforth, we call the difference between these two empirical AUCs the *regret*, which means that the regret of the best classifier is zero. Repeating this procedure 2000 times, we report the average regret per data set, learning algorithm, and AUC metric. In total, we experimented with classifiers from three learning algorithms: unpruned decision tree with Laplace correction (henceforth referred to as J48), naïve Bayes with kernel density estimation, and logistic regression.

The results are given in Table 3.2 and they clearly show that no AUC metric is able to consistently outperform any of the other metrics, regardless of the learning algorithm. In fact, the differences between the regrets are very small throughout. Moreover, from the regrets alone, it is impossible to conclude whether the differences between the metrics are due to small variations across the multiple runs or large differences in a few runs that represent a situation in which one of the metrics is clearly favoured. In Fig. 3.4, we therefore present the win-loss-equal statistics for each combination of two AUC metrics, as gathered over the 2000 runs for logistic regression (similar results were obtained for the other learning algorithms). These statistics are encoded as a horizontal bar chart for each data set, where the length of the bar to the left (right) of the baseline represents the fraction of wins (losses) for each metric combination. As can be seen, softAUC and probAUC often perform on par with AUC and thus often select the same model. Regarding the comparison of AUC and sAUC, the results are rather diverse and do not provide a clear picture.

To explain these results, we have observed that most of the time a small number of $k \ll 10$ candidate classifiers have a similar validation AUC value while the rest is significantly worse. This finding is not surprising given the setup of the experiments, namely, a classifier achieves a good performance only if it is trained on the features that are important for class separability. Given this, the small differences in the average regrets can be attributed to the different bias of the metrics. More specifically, all metrics select one of the top-k classifiers with a high probability. Yet, while AUC chooses the top-1 model with probability 1, and softAUC and probAUC are likely to select the same model due to their relatively small bias, the larger bias of sAUC causes it to make a selection in a more or less random way. This, however, is actually not a disadvantage: due to their almost equal performances, each of the top-k classifiers has more or less the same chance to achieve the best AUC on the test set, which explains that sAUC is indeed competitive.

[5] For completeness, we mention that a stratified split is a division of a data set into two parts such that the class distribution in both parts is identical to the class distribution of the data set as a whole.

Table 3.2: The average regret in the first setting for each data set and AUC metric, grouped by learning algorithm. For each of the three learning algorithms, the first column shows the regrets for AUC, the second for sAUC, the third for softAUC, and the fourth column shows the regrets for probAUC.

Data set	J48				NB				Logistic			
1	.0580	.0774	.0759	.0745	.0364	.0237	.0345	.0371	.0458	.0488	.0455	.0459
2	.0089	.0084	.0130	.0133	.0074	.0096	.0095	.0086	.0100	.0136	.0107	.0103
3	.0273	.0269	.0279	.0273	.0116	.0100	.0113	.0116	.0144	.0128	.0140	.0145
4	.0322	.0318	.0337	.0347	.0202	.0150	.0204	.0211	.0219	.0200	.0221	.0224
5	.0187	.0139	.0183	.0201	.0099	.0093	.0095	.0098	.0218	.0251	.0256	.0259
6	.0033	.0021	.0045	.0055	.0035	.0041	.0040	.0045	.0083	.0072	.0106	.0114
7	.0088	.0168	.0154	.0153	.0040	.0035	.0038	.0038	.0406	.0416	.0411	.0410
8	.0238	.2014	.1898	.1890	.0462	.0432	.0469	.0471	.0472	.0479	.0462	.0461
9	.0138	.0086	.0103	.0126	.0154	.0157	.0155	.0152	.0161	.0167	.0154	.0158
10	.0358	.1496	.1381	.1350	.0379	.0446	.0393	.0394	.0378	.0437	.0405	.0403
11	.0019	.0034	.0031	.0035	.0041	.0042	.0040	.0042	.0034	.0032	.0033	.0033
12	.0198	.0547	.0507	.0514	.0140	.0114	.0117	.0134	.0159	.0074	.0118	.0150
13	.0209	.0235	.0235	.0229	.0106	.0104	.0103	.0100	.0284	.0293	.0293	.0290
14	.0378	.0417	.0416	.0408	.0144	.0119	.0139	.0151	.0252	.0248	.0249	.0247
15	.0198	.0173	.0197	.0199	.0190	.0251	.0186	.0191	.0186	.0102	.0168	.0179
16	.0037	.0040	.0041	.0042	.0030	.0028	.0030	.0032	.0021	.0020	.0023	.0023

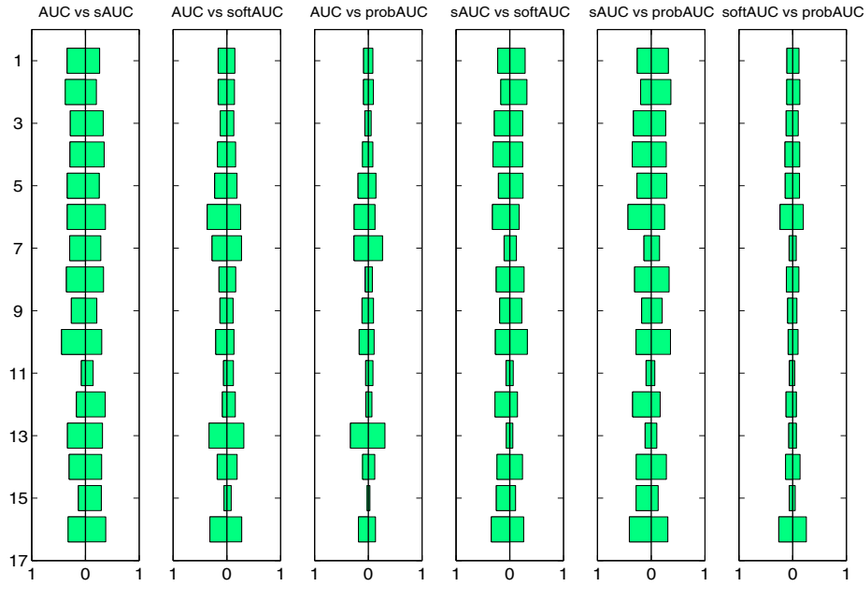


Figure 3.4: Win-Loss-Equal statistics for the AUC metrics in the first setting. The length of the left (right) bar of each combination of two metrics represents the fraction of wins (losses), and the fraction of equals is given by one minus the total length of the bars.

Besides, it is important to note that the experimental setup only compares classifiers of the same type. Our discussion in Subsection 3.4.1 and the example given there suggest that this is again favourable for sAUC. Indeed, we have argued that sAUC is expected to have problems when it comes to comparing classifiers producing scores with a different distribution or in a different range. In practice, this is often the case, for example when having to decide between classifiers from different learning algorithms such as a decision tree and a naïve Bayes classifier. Even when remaining within the same model class, changing the parameter(s) can have a large influence on the distribution of scores. Next, we analyse an example of that kind.

3.5.3 Setting 2: Changing the Learning Parameters

For the second setting, we construct a training set, validation set, and test set as in the first setting. Two different classifiers are then learned by applying J48 with and without Laplace correction, respectively, and no pruning. Laplace correction adds a pseudo-count of 1 to the class frequencies in the leafs, and therefore, shifts the scores away from the extreme values 0 and 1 more toward the middle. Trees with Laplace correction in the leafs have been shown to systematically outperform trees

Table 3.3: The average regret in the second setting for each data set and AUC metric. The first column shows the regrets for AUC, the second for sAUC, the third for softAUC, and the fourth column shows the regrets for probAUC.

Data set	J48 with different settings			
1	.0132	.0264	.0143	.0144
2	.0018	.0559	.0018	.0016
3	.0022	.0712	.0044	.0033
4	.0077	.0577	.0081	.0081
5	.0072	.0227	.0076	.0073
6	.0018	.0093	.0039	.0022
7	.0064	.0344	.0073	.0065
8	.0081	.0201	.0092	.0092
9	.0025	.0066	.0029	.0026
10	.0062	.0188	.0075	.0070
11	.0011	.0031	.0017	.0012
12	.0032	.0308	.0031	.0035
13	.0113	.0430	.0121	.0121
14	.0086	.0408	.0100	.0096
15	.0013	.0397	.0026	.0019
16	.0036	.0292	.0025	.0035

without Laplace correction; see for Zhang and Su (2006) and references therein. In the next chapter we will corroborate on these results in more detail.

In this experimental setting, it is expected that sAUC will select more often the unpruned tree without Laplace correction which, on average, has lower test AUC value. We do not expect large differences between softAUC and probAUC, although they should select better (worse) models than sAUC (AUC) does.

The average regrets of the metrics are reported in Table 3.3. The results confirm our expectations. The conventional AUC has lowest average regret in all but one data set. Also, there is no clear distinction between softAUC and probAUC, and most of the time both metrics choose the same model as AUC does. The sAUC metric performs significantly worse in all data sets since it is misled by the extreme scores produced by the unpruned trees without Laplace correction. The win-loss-equal statistics depicted in Fig. 3.5 give additional details. Clearly, all metrics except for sAUC often choose the same (best) model.

3.5.4 Conclusions from the Experimental Results

Our analysis of the experimental results confirms the conclusions that we derived from the theoretical analysis and the simulation studies. The results on the sixteen real data sets show that the conventional AUC metric is most effective for model selection. More specifically, dependent on the model selection scenario, the AUC is competitive with its variants or it clearly outperforms the variants.

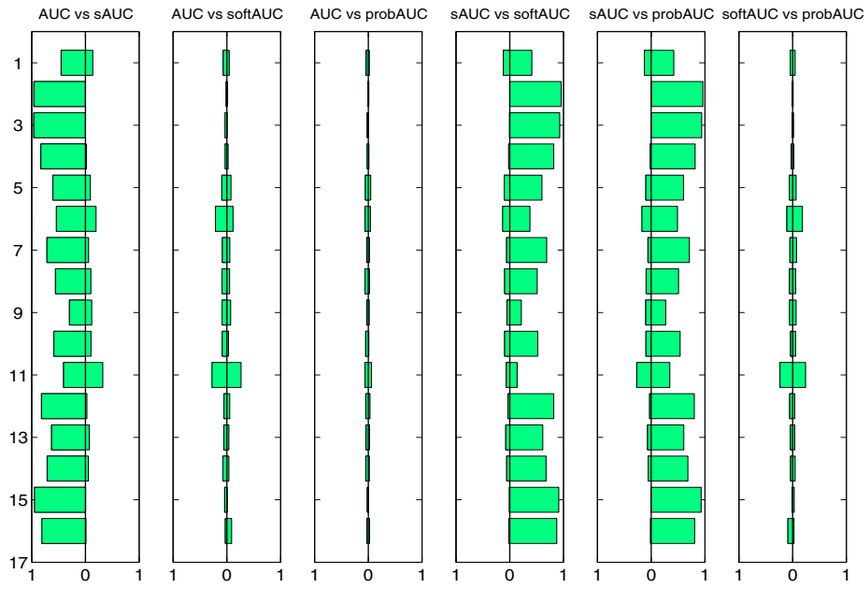


Figure 3.5: Win-Loss-Equal statistics for the AUC metrics in the second setting. The length of the left (right) bar of each combination of two metrics represents the fraction of wins (losses), and the fraction of equals is given by one minus the total length of the bars.

3.6 Implications for AUC-Optimising Classifiers

So far, we focussed on using the AUC for model selection. The metric is typically not used as an optimisation criterion in the learning phase of a classifier. Often, learning is aimed to optimise the error rate, cross-entropy, or mean squared error. Optimising with respect to these metrics does not, however, guarantee a model with maximum AUC (Yan, Dodier, Mozer, and Wolniewicz, 2003; Cortes and Mohri, 2003; Tax and Veenman, 2005; Caruana and Niculescu-Mizil, 2006). Since ranking the instances has become more important than only classifying the instances in some application domains (including law enforcement; recall Chapter 1), there has recently been a considerable interest in learning classifiers that maximise the AUC. We discuss two main research directions in learning AUC-optimising classifiers and we show how our results shed new light on this research and its results so far.[6] These directions are: convex optimisation problems (3.6.1) and gradient descent routines (3.6.2).

[6] Besides the two main research directions we will discuss, there has been an increasing interest in learning decision trees to maximise the AUC. We go into details in the next chapter.

3.6.1 Convex Optimisation

The first research direction is the formulation of an objective function and constraints for its optimisation. We note that directly optimising the AUC is hard since it is non-differentiable and many solutions may exist (e.g., consider the situation where the data is linearly separable). For these reasons, a regularised convex optimisation problem has been proposed (Rakotomamonjy, 2004; Brefeld and Scheffer, 2005; Tax, Duin, and Arzhaeva, 2006):

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2}\left\|\boldsymbol{w}\right\|^2 + C\sum_{i=1}^{P}\sum_{j=1}^{N}\xi_{ij} \\ \text{subject to} \quad & f(\boldsymbol{x}_i^p) - f(\boldsymbol{x}_j^n) \geq 1 - \xi_{ij} \ , \\ & \xi_{ij} \geq 0 \end{aligned}$$

where $f(x) = \langle \boldsymbol{w}, \boldsymbol{x} \rangle + b$, the constant C is the regularisation parameter, and the ξ_{ij} denote the slack variables. We note that $f(\boldsymbol{x}) \notin [0,1]$, but the scores can be rescaled. Introducing the corresponding dual problem shows that the instances only occur as inner products, and therefore a support vector machine (SVM) is obtained that approximately optimises the AUC. Despite the high computation time of this AUC-SVM, experimental results do not show consistent increases in test AUC. A first reason has recently been given by Steck (2007), where it was shown that minimising the hinge loss is an accurate approximation to maximising the AUC. This implies that an SVM already has high AUC. A second reason can be given from the results in this chapter, namely a slack variable $\xi_{ij} = 1$ only when $f(\boldsymbol{x}_i^p) = f(\boldsymbol{x}_j^n)$ and $\xi_{ij} > 1$ when the PN-pair is incorrectly ordered. The larger the absolute value of the score margin of discordant pairs, the higher the slack. Thus, the mapping from PN-pairs to slacks is similar to the sAUC modifier function (except for a shifting and reflection, as it serves as a penalty function). As we have seen, however, optimising sAUC is clearly different from optimising AUC. We therefore presume that, for the same reason, the approach may fail as an AUC-maximiser.

3.6.2 Gradient Descent

The second research direction is based on gradient descent routines. To make such routines applicable, the AUC needs to be approximated using a function that is continuous and differentiable. A popular choice is a steep sigmoid function. Therefore, the proposed algorithms learn classifiers that optimise the softAUC, and indeed it has been verified that these classifiers have higher test AUCs than those obtained by AUC-SVMs (Herschtal and Raskutti, 2004; Calders and Jaroszewicz, 2007). Also, it has been observed by Calders and Jaroszewicz (2007) that significantly better test AUCs are obtained by increasing β in (3.4). Our results give a well-founded explanation for these findings. Of course, a disadvantage of the gradient descent routines is that they are restricted to learning hyperplanes in input space. It would therefore be an interesting direction of future research to extend AUC-SVMs to incorporate proper modifier functions, although improvements are likely to be small.

3.7 Related Work

We have considered a model selection scenario where the goal is to select the model with highest AUC and the selection is based on the AUC or one of its variants.

In general, one may speak of a *goal metric* and a *selection metric*. We have seen that, when the goal metric is the AUC, the best selection metric among the AUC and its variants is again the AUC (despite that it has been argued as less robust in case of small score margins). Although we are the first to show this result, there has been some work regarding different goal metrics and/or selection metrics that are worth mentioning in this chapter.

Several research papers focus on the setting where the goal metric and the selection metric is the AUC. The interest is whether the selected model is also statistically significantly better than the other candidate models; see the works by McNeil and Hanley (1984), Mason and Graham (2002), Obuchowski *et al.* (2004) and references therein. Parametric and non-parametric tests have been proposed, although the assumptions may not always be valid in practice. Several issues have to be recognised in order to choose the best statistical test.

Rosset (2004) is interested in the correlation between error rate and AUC. More specifically, he compared these two metrics in a model selection scenario where the goal metric is the error rate. Surprisingly, he showed that model selection via the AUC often leads to better models, despite that his results should be considered with reservation since the experimental setup only involves very few data sets and only two classifiers. Nonetheless, other works by Huang and Ling (2006), Caruana and Niculescu-Mizil (2006), Skalak *et al.* (2007) claim to have results in the same trend, while Cortes and Mohri (2003) show formally that AUC-maximisation is different from minimising the error rate, especially when the error rate is low. However, the assumption of their formal analysis is too strong since all rankings with t errors are assumed equiprobable.

Recently, some of the aforementioned experiments were repeated by Huang, Ling, Zhang, and Matwin (2008) except that a significance test is added to verify whether different models preferred by the selections metrics are also statistically different. Excluding results for which the test showed no significant difference, the best results are obtained when the goal metric and the selection metric are identical.

3.8 Chapter Conclusions

In this chapter, we focused on RQ 1: *To what extent is the area under the ROC curve an effective performance metric for ranking when compared to its variants that consider the absolute values of the scores?* The variants (enhancements of the AUC) have been proposed since at first sight it may appear unreasonable for a ranking performance metric to ignore the absolute value of the score margin of PN-pairs. So, the variants are less favourable toward models that generate small score margins and instead prefer models producing large margins, as small margins are considered as uncertain, and therefore may contribute to the AUC just by chance.

To answer our RQ 1, we presented a general framework that allows for a unified treatment of the conventional AUC metric and its variants. A formal analysis addressed the bias and variance of the estimates of the true AUC value. The results of this analysis are supported by strong empirical evidence, which leads us to conjecture that none of the variants is as effective as the AUC itself. Hence, a clear and well-founded answer to RQ 1 has been formulated.

More specifically, based on the findings of this work, we may draw the following three conclusions. First, the three AUC variants are all biased and their variance can go in either direction. The net effect on the quality of the estimations is thus not clear and, hereby, there is no solid theoretical foundation for the variants. Second, our empirical results have shown that the conventional AUC cannot be outperformed systematically, not in an ideal setting according to the theoretical analysis, and not in real model selection scenarios. The variants with a modifier function that closely resembles the step function perform best. In this respect we can also see that softAUC and probAUC, with properly chosen parameters, are accurate approximations of the conventional AUC metric. Third, learning AUC-optimising classifiers works best with a symmetric modifier function that sharply smoothes out the region around zero score margin.

Chapter 4

AUC-Optimising Decision Trees

Decision trees are known to be good classifiers but less good rankers. A few methods have been proposed to improve their performance in terms of AUC, but there does not exist a formal analysis or a unified explanation of the success of these methods. In this chapter, we show that the number of distinct scores produced by a decision tree is crucial for its AUC. Several implications are discussed, along with several experiments. Our result provides an answer to the second research question.[1]

4.1 Introduction

Decision trees are one of the most extensively studied methods in machine learning and data mining. Several factors contribute to their popularity, notably the following. Decision trees are comprehensible and interpretable by the domain expert and can naturally handle different types of features (e.g., numerical and categorical). Moreover, they are intrinsic multi-class learners, scale comparatively well with the number of features and instances (Provost and Kolluri, 1999), and have a feature selection mechanism built-in. Finally, decision trees have shown to obtain classification performance close to or even outperforming state-of-the-art methods, especially when they are boosted or used in a different ensemble method (Geurts, Ernst, and Wehenkel, 2006). The two most widely used implementations for decision trees are CART (Breiman, Friedman, Stone, and Olshen, 1984) and C4.5 (Quinlan, 1993).

From a practical point of view, motivated by possible applications in the field of law enforcement, we believe that the balance between comprehensibility and good classification performance of decision trees is of utmost importance. As an example, we consider classifying offenders into having a high risk or a low risk for recidivism. An offender has as a high risk when he turns out to become a moderate-rate desister or a high-rate desister. The interested reader can consider Blokland *et al.* (2005b) for definitions of these terms; for now, it is sufficient to have an intuitive meaning,

[1] This chapter is based on an article by Hüllermeier and Vanderlooy (2009b) which extends earlier work in Hüllermeier and Vanderlooy (2008a).

namely that the high-risk class represents offenders that can be considered as severe recidivists. The training data used to induce a decision tree consists of offenders that are characterised by features such as number of previous convictions, age of first conviction, alcohol addiction, type of occupation, gender, living in big cities or rural areas, and so on. These data are recorded in a fixed time period and we use a decision tree to make a prediction at the start of the time period. The learned decision tree is shown in Fig. 4.1.[2] The tree has a correct prediction rate of ≈ 0.78 (i.e., its classification accuracy is 78%) and it is interpretable in the sense that a path from the root to a leaf can be considered as a conjunction of feature tests in combination with a final class label. This leads to classification rules, e.g., following the rightmost path, we have the following rule: "if the offender has more than 4 convictions in a small time period and if he commits crimes in a big city, then he is likely to be a severe recidivist". Moreover, the decision tree automatically shows which features are most important since features tested close to the root are better in distinguishing low-risk from high-risk offenders than features lower in the tree.

A decision tree, trained in the usual way as a discrete classifier, can be used for ranking by scoring an instance in terms of the frequency of positive training examples found in the leaf to which the instance is assigned. For example, an instance belonging to a leaf with 10 positive examples and 400 negative examples will receive a score of $10/410 \approx 0.025$, indicating that the instance is likely to be a negative one. The score is thus an estimate of the true conditional probability of the positive class. However, several experimental results indicate that decision trees perform poorly in ranking instances, although the definition of the score is clearly reasonable. Therefore, in this chapter, we are interested in answering our second research question (RQ 2): *How can the AUC of an interpretable and comprehensible classifier, such as decision trees, be optimised?*

There have been some methods proposed to improve the AUC of decision trees, along with first empirical evidence of their effectiveness. Two papers provide experiments showing that unpruned trees lead to better rankings, i.e., higher AUC values, than standard pruned trees (Provost and Domingos, 2003; Ferri, Flach, and Hernández-Orallo, 2003). This is counterintuitive at first sight since it is well-known, at least for classification accuracy, that pruning prevents or alleviates the overfitting effect and strongly reduces variance. A third paper claims to obtain even higher AUC values with a "soft" decision tree in which an instance can be assigned to several leaves simultaneously (Ling *et al.*, 2003). Yet again, no formal analysis of the method and its ability to improve the AUC was presented.

In this chapter we first replicate and extend the previous empirical studies, not only to improve our understanding of the earlier results, but also to explain, correct, or refine some implicit assumptions and conjectures that have been made by several authors. Second, for the first time, we present a formal analysis about AUC-optimising decision trees. We show that the AUC is likely to increase with the number of scores produced by a tree, at least when these scores are reasonable (i.e.,

[2] The decision tree is learned on the data collected by Blokland (2005a). We eliminated some features to keep the depth of the tree of reasonable size. The tree serves only as an illustration. So, by no means, the tree should be interpreted from a true practice viewpoint.

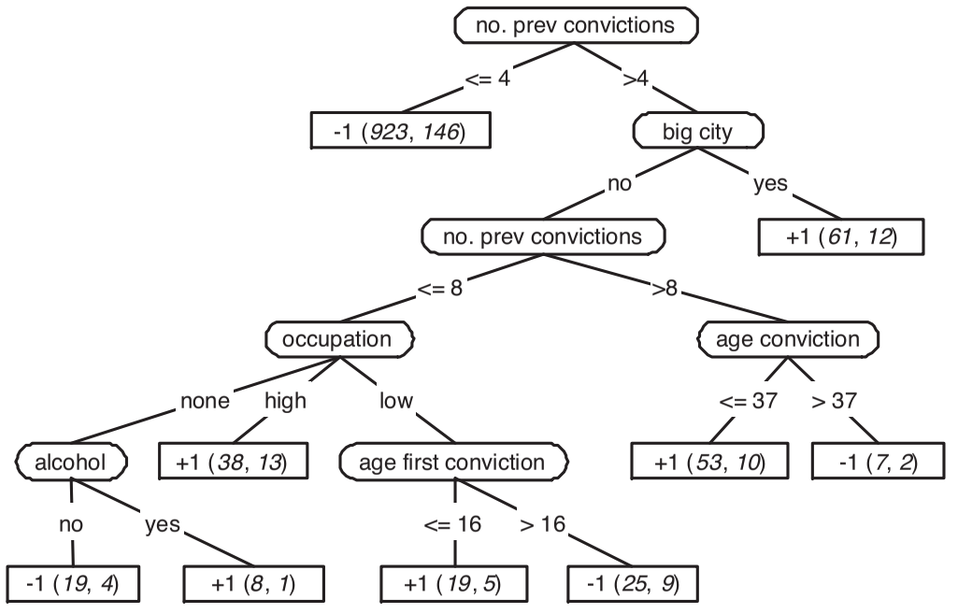


Figure 4.1: An example of a decision tree that predicts if an offender has a high risk (class label +1) or low risk for recidivism (class label -1). The two numbers between brackets in the leaves denote the number of instances falling in the leaf and the number of incorrect classifications made by the leaf (on the training data).

better than random) approximations of the true conditional probabilities in the leaves. Simulation studies with synthetic data verify our formal analysis and show the robustness toward estimation errors. Finally, as a byproduct, our result suggests a simple method for improving the ranking performance of decision trees, and it provides an understanding why some classifiers are better rankers than other classifiers. Hence, our answer to the research question generalises to classifiers different from decision trees.

The remainder of the chapter is organised as follows. Decision trees are discussed in Section 4.2. In Section 4.3 we review related work and in Section 4.4 we provide a first set of experiments that verify, correct, and extend the earlier empirical results. Inspired by the outcome, we provide our formal analysis in Section 4.5. In Section 4.6 we present and experiment with an intuitive method to improve the AUC of decision trees. In Section 4.7 we provide a short discussion on the obtained results and their generalisation to other classifiers. Finally, Section 4.8 concludes the chapter and provides our answer to the research question.

4.2 Decision Trees as Scoring Classifiers

As in the previous chapter, we consider a supervised learning scenario where one is interested in learning an accurate scoring classifier. Background and notation introduced there is also used in this chapter. In addition, in the next two subsections we provide extra information by briefly discussing decision tree learning (4.2.1) and smoothing scores with Laplace correction (4.2.2).

4.2.1 Decision Tree Learning

A decision tree is often learned as a discrete classifier and afterward transformed to a scoring classifier. The learning strategy is a recursive one and is known as divide-and-conquer. First, a feature is selected to be tested at the root of the tree. This splits the sample of training examples in different subsets, one subset for every value (or range of values) of the feature. Then, the same procedure is applied to each branch, using only the examples that reach that branch. There are various methods to decide which feature is best to split although they all perform alike. We refer the interested reader to the work by Esposito, Malerba, and Semeraro (1997) and references therein. Splitting can stop for various reasons, e.g., when no more features can be tested on the branch, when all examples have the same class label, or when a minimum number of examples remains.

The learning strategy will almost always lead to decision trees that overfit the training data set. For this reason, the complexity (depth) of the tree is reduced by *pruning*, i.e., by removing subtrees and replacing them by leaves.[3] The pruning algorithm that we used in our experiments is the standard one used in C4.5 and works

[3]Besides this pruning operation, it is also possible that a subtree is *raised*. This means that a subtree is placed one level closer to the root, hereby replacing its parent node. We do not go into details here since it is not important for the remainder of the chapter.

as follows (Witten and Frank, 2005, pp. 193-196). Consider a leaf with t examples of which i do not belong to the most frequent class. An estimate of the error rate is i/t although this estimate may be optimistic. So instead, we assume that the t examples are generated by a Bernoulli process with parameter q (the true probability of an error), out of which m examples appeared to be errors. Then, a pessimistic estimate of the error rate is computed as the upper limit of the binomial probability when i "successes" have been observed in t randomly drawn examples. A user-specified confidence level should be provided for this calculation. For a subtree, the estimated errors of the branches are weighted according to the number of examples they cover and then summed. A subtree is replaced by a leaf when the error rate of the former is at least that of the latter.

A decision tree is a scoring classifier when the scores are considered as the relative frequencies of the positive class in the leaves. Hence, the number of leaves is an upper bound on the number of different scores the tree can produce. The ROC convex hull of a decision tree can be generated efficiently (Ferri, Flach, and Hernández-Orallo, 2002). To see this, we note that we can identify $i+1$ ranges of thresholds when there are i different scores assigned to instances in a sample S. Each thresholding range results in a different classification performance measured in terms of true positive rate, *tpr*, and false positive rate, *fpr*. Hence, each leaf corresponds to a line segment of the curve, with slope equal to the class distribution in that leaf. Ordering the leaves according to their class distributions (i.e., the scores they produce) generates a ROC convex hull. So, the more leaves, the more line segments we can have.

In the remainder of this chapter, when we consider ROC curves, we always assume them without restriction to be ROC convex hulls. Also, we adopt a geometric perspective in the formal parts, and therefore the AUC is defined as the sum of the area of the trapezoids defined by the line segments of the convex hull.

4.2.2 Laplace Correction

The so-called Laplace correction is often used in probability estimation. It simply adds a pseudo-count of 1 to the relative frequencies of all classes. For a decision tree, this means that, given p positive examples and n negative examples in a leaf, the estimated probability for the positive class is defined as

$$\frac{p+1}{p+n+2} \ ,$$

instead of the observed class frequency $p/(p+n)$. We note that this estimated probability is the score in our context. Roughly speaking, the Laplace correction yields a more cautious probability estimation by *smoothing* the scores in the sense that they are shifted from the extremes 0 and 1 toward the indifference probability value 0.5. In fact, neither 0 nor 1 can be produced as a score, although convergence toward these values is possible for large sample sizes.

Even though Laplace correction appears to be ad-hoc at first sight, it can be derived from a uniform prior on [0, 1] as a Bayes estimate of the success parameter of a binomial distribution (Bishop, 2007, pp. 71-74).

4.3 Related Work

The leaves of a decision tree have been shown to produce poor scores (Smyth, Gray, and Fayyad, 1995; Kohavi, 1996; Zadrozny and Elkan, 2001). So, it seems that the learning strategy of decision trees is not suitable for learning good rankers. Several researchers have tried to improve the ranking performance ever since (Ferri *et al.*, 2002; Ferri *et al.*, 2003; Wang and Zhang, 2006; Zhang and Su, 2006; Alvarez, Bernard, and Deffuant, 2007; Clémençon and Vayatis, 2008).

In this chapter we restrict ourselves to improvements involving a single decision tree and, therefore, do not consider methods such as bagging since it produces incomprehensible results. Of special interest to us is the paper by Provost and Domingos (2003) in which two enhancements are proposed to improve the AUC of decision trees. The authors call the resulting method C4.4 and show empirically that it leads to higher test AUC values than C4.5 (the standard learning algorithm that has been used as a baseline). The two proposed enhancements are the following.

- First, learn unpruned trees by turning off error-based pruning and node collapsing. The stopping criterion for tree building is then that either the number of instances in the leaf is at most two, or all the features have been exhausted on the current path. The main idea is that the branches removed by the pruning step might still be useful for ranking. They allow for assigning different scores to instances contained in a hyper-rectangle that is considered good enough for classification purposes.[4] A drawback of this technique is that a decision tree grows exponentially in the number of internal nodes.

- Second, smooth the class frequencies in a leaf using Laplace correction. An unpruned tree will often produce leaves with low *cardinality* (i.e., the leaves comprise only a small number of training instances) and, therefore, the tree will produce unreliable scores. The most compelling example is a pure leaf, which is a leaf consisting of only positive or only negative examples. Such a leaf produces a score of either 0 or 1, indicating that it is most certain of its output, even if the pure leaf has low cardinality. Laplace correction makes the scores from small leaves less extreme and, as conjectured by the authors, hereby more accurate and reliable. The win-loss-equal statistics that they obtained show that most of the improvement of C4.4 over C4.5 is due to the Laplace correction. Interestingly, its more general form, called m-estimate, was found to not produce higher test AUC values (Ferri *et al.*, 2003).[5]

In a response to these frequently cited empirical results, Ling *et al.* (2003) mention the following two possible disadvantages of C4.4. First, the unpruned tree will probably overfit and this may cause unreliable class frequencies in the leaves. Second, the number of training examples in the leaves of an unpruned tree is often

[4] We note that a decision tree divides the input space by means of hyper-rectangles since, at each node, a feature test induces a decision boundary perpendicular to the corresponding axis.

[5] We note that Laplace correction does not change the accuracy of a decision tree, except when the score is 0.5 (which is unlikely due to the tree learning strategy).

(very) small. It follows that these leaves are likely to give the same score to test instances, yet these instances may be quite different from each other since they follow different branches in the tree. To overcome these possible disadvantages, the authors propose the following method to improve the ranking performance. The tree building strategy is left unchanged, i.e., the learning algorithm C4.5 is still used. However, in the prediction phase, a test instance is propagated along all branches emerging from a node, with a smaller weight for the branches not satisfying the test predicate in that node. The weights are multiplied along the paths toward leaves and corresponding predictions are averaged. So, each leaf contributes to the overall score of a test instance and the weight of the contribution is determined by the number of feature tests that failed on the path from root to leaf. As a rationale of this approach, the authors explain that it is natural to expect a small chance that feature values are altered due to noise or errors in the training set. Hence, different paths in the tree should be explored. Experimental results show that this technique is able to outperform C4.4 when the pruning level and a second parameter controlling the weight of each branch are set appropriately. Yet, only six small data sets are used in these experiments.

4.4 Experimental Analysis of AUC-Optimising Trees

In this section, we present a first set of experiments that elaborate on some details and (implicit) assumptions of the aforementioned two methods. After presenting the experimental setup (4.4.1), we investigate the effect of three factors that have been considered to be influential for the AUC of decision trees, namely: the pruning level (4.4.2), Laplace correction (4.4.3), and the number of distinct scores (4.4.4). We end this section with a conclusion from our empirical findings (4.4.5).

4.4.1 Experimental Setup

We used twenty binary classification data sets from the UCI benchmark repository (Asuncion and Newman, 2007). The data sets vary strongly in size, number and type of features, and in class distribution. Their most important characteristics are given in Table 4.1. The tree induction algorithm that we use is the implementation of C4.5 provided by the `WEKA` machine learning package (Witten and Frank, 2005).

The reported values of the AUC (or any other test statistic to our interest) are obtained as follows. We apply a 10-fold stratified cross-validation procedure and on each training fold we learn ten decision trees with different pruning levels. The pruning level is controlled by the *confidence factor* in the learning algorithm in the sense that smaller confidence factors incur more pruning. The confidence factor of decision tree no. $i \in \{1, \ldots, 9\}$ is set equal to $i/20$, and the tenth decision tree is a completely unpruned tree (pruning turned off, confidence factor is irrelevant).[6] For

[6] The default value of the confidence factor is 0.25, so a standard pruned tree (i.e., a conventional decision tree) in earlier experiments is our decision tree no. 5. We also note that the confidence factor should always be set between 0 and 0.5.

Table 4.1: The twenty data sets, where the column headings refer to: (1) reference number, (2) data set name, (3) number of instances, (4) number of nominal features, (5) number of numerical features, and (6) percentage of the majority class.

#	name	size	nom	num	% maj class
1	adult	48842	7	6	76.07
2	breast cancer	286	9	0	70.28
3	breast wisconsin	699	0	8	65.52
4	caravan	5822	85	0	65.52
5	credit rating	690	9	6	55.51
6	german credit	1000	13	7	69.40
7	heart statlog	270	7	6	59.50
8	horse colic	368	14	7	63.04
9	house votes	435	16	0	38.62
10	ionosphere	351	0	34	35.90
11	kr vs kp	3196	36	0	52.22
12	liver	345	1	5	42.03
13	monks1	556	6	0	50.00
14	monks2	604	6	0	65.72
15	pima	768	0	8	65.10
16	sick	3772	22	7	93.16
17	sonar	208	0	60	53.36
18	spambase	4601	0	57	61.00
19	spect	267	22	0	58.80
20	tic-tac-toe	958	9	0	65.34

each pruning level, the values of the test statistic on the ten test sets are recorded and averaged. This procedure is repeated twenty times and each time the data set is randomly shuffled. We report the final mean values of the test statistic and the standard deviation of its mean.

In order to analyse some of the results more thoroughly, we applied the Wilcoxon signed-ranks test to detect statistically significant differences as recommended by Demšar (2006). The test looks at the difference between the test statistic values of two methods on each data set. These differences are ranked according to their absolute values; average ranks are assigned in case of ties. The test statistic is then given by the difference between the sum of ranks for the data sets on which the second method outperformed the first and the sum of ranks for the opposite case.

4.4.2 Dependence of AUC on Pruning Level

Unpruned trees with Laplace correction have been shown to consistently outperform standard pruned trees in terms of AUC (Provost and Domingos, 2003). However, the effect of decreasing the pruning level has not been tested and it was implicitly assumed to be monotone. Also, no tests have been performed comparing different

pruning levels for a decision tree without Laplace correction. So, to complement earlier experiments, we show in Fig. 4.2(a) the typical behavior of the AUC obtained by the ten decision trees for four representative data sets.[7] The pruning level is decreasing from left to right on the horizontal axis, i.e., left is highly pruned and right is completely unpruned. Solid and dashed curves show the result for decision trees with and without Laplace correction, respectively. From these illustrations, we can make three interesting observations.

- First, comparing dashed and solid curves, we have that Laplace correction applied in the leaf nodes never decreases the AUC, and hence, the Laplace-corrected decision trees significantly outperform conventional trees. We thus strongly advocate to *always* use Laplace correction.

- Second, in conjunction with Laplace correction, unpruned trees indeed produce better rankings than the trees pruned at the default level. The Wilcoxon signed-ranks test shows that we can reject the null hypothesis (i.e., the ranking performance of unpruned trees and standard pruned trees are equally well) at the 1% significance level. However, we note that the value of the AUC is not always a monotone decreasing function of the pruning level, or equivalently, a monotone increasing function of the depth of the tree. Therefore, it seems to be a good idea to try out several pruning levels below the standard value used for classification purposes.

- Third, no clear trend can be identified when inspecting the dashed curves only (trees without Laplace correction) although, in general, the standard pruned trees have higher AUC values than unpruned trees. Nonetheless, when using the Wilcoxon signed-ranks test, we do not have a statistically significant difference at the 5% significance level due to some data sets where the unpruned tree outperforms with a large margin the standard pruned tree.

Table 4.2 summarises these three findings for all data sets by reporting the AUC of decision trees no. 1, 5, and 10. There is a part in the table for the results with and without Laplace correction, respectively, and a fourth column indicates the pruning level where the highest AUC value occurs. In this way, we still have information concerning all ten decision trees used in each part. By inspecting the test statistics, we see that the results verify that the best rankers are decision trees with Laplace correction and, albeit a few exceptions, with much less pruning than is commonly used for classification purposes. Finally, we remark that there were only 4 out of 2000 results (20×10 fold cross-validation with 10 trees) for which the corresponding decision tree obtained a lower AUC when Laplace correction was applied. Nonetheless, the cases occurred for small trees that were far from having a maximum AUC, and moreover, the differences were extremely small.

[7] These data sets are representative in the sense that the shape of the curves of the test statistics and the corresponding differences in absolute values represent the other data sets as well.

4.4.3 Effect of Laplace Correction

The above results show that Laplace correction has an important effect; in fact, it yields the largest improvement in AUC. In earlier work, this effect was attributed to an increased reliability, i.e., scores of small leaves are presumably less reliable estimates of the true positive class probabilities as are scores of large leaves. This distinction is especially considered to be important when extreme scores are produced since corresponding instances are ranked at the very ends of the ordered list of instances, and hereby, may incur large ranking errors.

More specifically, consider a leaf with p positive and n negative examples, and a second leaf with corresponding values p' and n'. Moreover, consider that both have the same score $s_1 = p/(p+n) = p'/(p'+n') = s_2$. The corresponding Laplace-corrected scores are $s'_1 = (p+1)/(p+n+2)$ and $s'_2 = (p'+1)/(p'+n'+2)$, respectively, which means that $s'_1 > s'_2$ if and only if $p + n > p' + n'$. In other words, Laplace correction breaks ties between leaves with the same score by preferring larger leaves to smaller ones. For example, an instance falling in a leaf with 40 positives and 10 negatives will be ranked ahead of an instance in a leaf with only 4 positives and 1 negative. From a statistical point of view, this is clearly reasonable since the prediction for the positive class is arguably more reliable for the former than for the latter leaf. Analogous corrections are made in the case where $p < n$, i.e., for presumably negative leaves.

It is worth noting that, apart from breaking ties between equally scored instances, Laplace correction may also result in a reversal of scores. Thus, it may happen that $s_1 < s_2$ while $s'_1 > s'_2$ for the corrected scores, a property that is perhaps less desirable. In particular, instances falling in leaves with only positive (negative) examples should still be ranked above (below) the instances falling in non-pure leaves. This is however not guaranteed by the Laplace correction, especially not in case of leaves with low cardinality (the score of a pure leaf with two positive instances, for example, would be strongly reduced from 1 to 3/4). To elaborate on this, we tested whether Laplace correction changes the rank position of extremely scored instances with respect to instances that do not fall into pure leaves. In our experiments (results shown in the next paragraph), we have found that this is unlikely to happen, and when it happens, then the change in rank position is on average very small. For this reason, we conjecture that the true effect of Laplace correction comes from its *local tie breaking*, which is especially relevant for scores assigned by small leaves. In other words, some equalities between the scores are turned into strict order relations and this happens mostly without swapping the rank position of the instances.

We do not report a table with empirical results for our conjecture, but instead show a scatter plot depicting purity of the leaves (i.e., frequency of the positive class) against cardinality; see Fig. 4.3. Data points in the scatter are gathered over all runs of the experiment. Pluses represent data points without Laplace correction, and we again show results for decision trees no. 1, 5, and 10. Two observations can be made. First, when the pruning level decreases, more extreme scores are made and the leaves become smaller. Second, Laplace correction leads to more diversity among the extreme scores, and changes in rank position are often limited to the

corresponding instances and its close ranking neighbours since we do not cross over many pluses that have purity less than one (especially not in a single test fold where there is much less data than shown in the figure). More insight in the second observation is obtained by inspecting Fig. 4.4. It shows the average fraction of non-pure leaves that a Laplace-corrected pure leaf "jumps" over. In other words, from a ranking point of view, the figure shows the average fraction of positions that a randomly chosen instance with extreme score moves from the end of the ranked list toward the middle due to Laplace correction. Clearly, this fraction is often small except for a few data sets, but here all ten decision trees generated extreme scores or scores very close toward the extremes (accounting for the bars in the right of the figure). Thus, in general, these results verify the local tie breaking effect.

4.4.4 Number of Distinct Scores

As a last experiment, we investigate a conjecture by Ling *et al.* (2003), namely that large trees are only able to produce a small number of different scores. Instances with an equal score are potentially disadvantageous for AUC because they can at most contribute a count of $1/2$. Hence, to obtain more diversity among the instances, a pruned tree should be preferred. The main observation here is that the cardinality of a leaf determines the number of scores it can produce, i.e., a leaf with t examples can only produce $t+1$ different (uncorrected) scores, namely $0, 1/t, 2/t, \ldots, 1$.

However, we conjecture that the net effect of pruning on the total number of scores produced by a tree is not clear since pruning indeed increases the cardinality of the leaves (and, therefore, reduces the probability that two leaves have the same score) but at the same time the number of leaves becomes smaller. Besides, it is clear that Laplace correction increases the number of scores since two leaves produce the same score only when the absolute numbers of positive and negative instances are identical (and not only the relative frequencies).

Figure 4.2(b) shows the typical behaviour of the number of different scores for the four representative data sets and Table 4.3 summarises the findings for all data sets and decision trees no. 1, 5, and 10. The presentation of these results is identical to the results about the AUC values that the trees have obtained (cf. Subsection 4.4.2). We make the following two observations from these illustrations and statistics, depending on the type of features in the data set.

- First, for the data sets with only numerical features, the number of scores is almost always non-decreasing in the pruning level (since a numerical feature can be tested repeatedly on a path). When we have a decrease, then this decrease is very small, often limited to low pruning levels, and only happens in case of no Laplace correction. This shows that a few leaves can indeed start to produce the same scores, but Laplace correction is able to break the ties.
- Second, even for data sets with only nominal features we can have an increase in scores along all pruning levels (e.g., `spect` and `house votes`). When this is not the case, we often see that the increase is up to a certain pruning level. After that level, trees with Laplace correction slightly decrease the number of

scores (this happened only on four data sets). Without Laplace correction the decrease is of course more strongly.

Independent of the aforementioned two observations, it is interesting to compare Figs. 4.2(a) and 4.2(b), suggesting that the decision trees that produce good rankings are the ones that assign many different scores to the test instances.

4.4.5 Conclusions from the Experimental Results

With this set of experiments we have verified earlier results in more detail and we tested some issues that have not been carefully considered before. Summarising the results, we confirm that Laplace correction is always beneficial and that unpruned trees in general have higher AUC than standard pruned trees (at least with Laplace correction). In addition, a slightly pruned tree can have higher AUC than an unpruned tree and this almost always when it produces more distinct scores. From these empirical findings, we may conclude that there seems to be a strong positive correlation between number of scores and the AUC value.

In the next section, we present a formal analysis that indeed shows that a higher number of distinct scores (implying less ties among ranked instances) is likely to result in a higher AUC.

4.5 Formal Analysis of AUC-Optimising Trees

In the previous section, we have seen among others that unpruned trees have higher AUC than pruned ones. To obtain a better idea of why this is the case, we provide a formal analysis showing that, under certain weak assumptions, tie breaking of scores by means of node splitting (i.e., keep on growing a decision tree) is beneficial in the sense that it can only increase the AUC. It is straightforward to extend this result to the other observations from our experiments, e.g., the effect of Laplace correction.

4.5.1 Tie Breaking Improves AUC

We will mainly adopt a geometric perspective as we believe that it greatly facilitates the understanding. Given a set of instances S, the ROC curve of a decision tree is constructed as follows. Let the leaves be numbered in decreasing order according to their score and denote by L_i the i-th leaf, or depending on the context, the set of (training) examples in the i-th leaf ($i = 1, \ldots, m$). With p_i and n_i denoting the number of positive and negative examples in L_i, respectively, the score of this leaf is $s_i = p_i/(p_i + n_i)$. Let $P = p_1 + \ldots + p_m$ be the total number of positive instances in S and, analogously, we define $N = n_1 + \ldots + n_m$. Now, each leaf L_i contributes a segment S_i with slope $d_i = (p_i \cdot N)/(n_i \cdot P)$, and lengths $\Delta x_i = n_i/N$ and $\Delta y_i = p_i/P$ in the directions of the abscissa and ordinate, respectively. The concatenation of these ordered segments leads to a piecewise linear convex curve, which is the ROC convex hull. In general, we denote the concatenation of segments S_a and S_b by $S_a|S_b$.

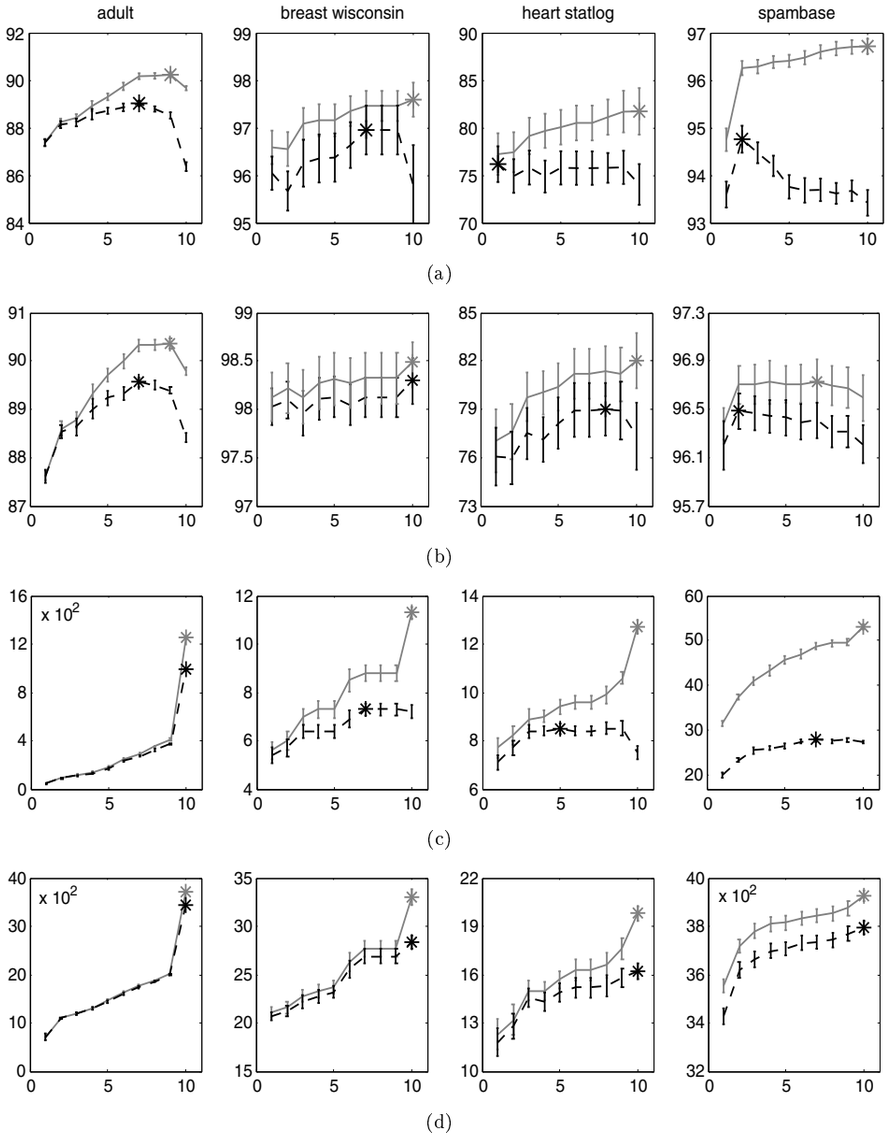


Figure 4.2: Test statistics of the ten decision trees. Solid and dashed curves represent trees, respectively, with and without Laplace correction. An asterisk indicates the first tree with the highest test statistic, which is the AUC of C4.5 and the perturbation method (a - b) and corresponding number of different scores (c - d).

Table 4.2: The average AUC and standard deviations of decision trees no. 1, 5, and 10. Results are shown in two parts: without and with Laplace correction. The fourth column of each part lists the decision trees that obtain the highest AUC.

Data set	Without Laplace correction				With Laplace correction			
	1	5	10	max	1	5	10	max
1	.8739 ± .0014	.8874 ± .0013	.8638 ± .0019	7	.8741 ± .0015	.8932 ± .0014	.8969 ± .0009	9
2	.5691 ± .0176	.5942 ± .0285	.5830 ± .0246	2-3	.5691 ± .0176	.5933 ± .0278	.6127 ± .0210	2-3
3	.9605 ± .0035	.9638 ± .0051	.9580 ± .0084	7-9	.9660 ± .0035	.9717 ± .0034	.9760 ± .0036	10
4	.5000 ± .0000	.4997 ± .0004	.5777 ± .0091	10	.5000 ± .0000	.4990 ± .0005	.6997 ± .0093	10
5	.8915 ± .0057	.8940 ± .0063	.8530 ± .0121	5-6	.8929 ± .0060	.9079 ± .0055	.9022 ± .0034	8
6	.6261 ± .0260	.6467 ± .0148	.6253 ± .0142	2	.6323 ± .0261	.7100 ± .0084	.7090 ± .0108	2
7	.7622 ± .0188	.7583 ± .0176	.7408 ± .0215	1	.7728 ± .0219	.8011 ± .0182	.8178 ± .0245	10
8	.8517 ± .0192	.8517 ± .0192	.8414 ± .0167	1-5	.8517 ± .0192	.8517 ± .0192	.8617 ± .0158	10
9	.9670 ± .0082	.9797 ± .0073	.9762 ± .0076	2	.9728 ± .0061	.9852 ± .0041	.9875 ± .0043	10
10	.8663 ± .0170	.8660 ± .0150	.8822 ± .0141	10	.8998 ± .0132	.9083 ± .0137	.9126 ± .0144	10
11	.9948 ± .0013	.9967 ± .0012	.9969 ± .0012	10	.9967 ± .0005	.9988 ± .0003	.9991 ± .0002	10
12	.5753 ± .0152	.6251 ± .0150	.6423 ± .0144	10	.5723 ± .0145	.6406 ± .0128	.6537 ± .0139	10
13	.9233 ± .0145	.9823 ± .0054	.9804 ± .0051	9	.9233 ± .0145	.9823 ± .0054	.9858 ± .0037	9
14	.5000 ± .0000	.5356 ± .0168	.6532 ± .0103	8	.5000 ± .0000	.5388 ± .0179	.6742 ± .0099	8
15	.7412 ± .0145	.7558 ± .0084	.7546 ± .0102	4	.7533 ± .0134	.7805 ± .0098	.7870 ± .0104	10
16	.9468 ± .0088	.9523 ± .0075	.9512 ± .0089	8-9	.9441 ± .0101	.9704 ± .0076	.9917 ± .0027	8-10
17	.7344 ± .0197	.7439 ± .0222	.7416 ± .0213	2-3	.7841 ± .0160	.7814 ± .0145	.7851 ± .0156	2-3
18	.9360 ± .0027	.9377 ± .0025	.9343 ± .0027	2	.9476 ± .0024	.9642 ± .0013	.9672 ± .0016	10
19	.7112 ± .0215	.6626 ± .0268	.7033 ± .0210	1	.7191 ± .0206	.7121 ± .0188	.7325 ± .0210	7
20	.8352 ± .0136	.9117 ± .0049	.8966 ± .0084	9	.8314 ± .0133	.9229 ± .0042	.9374 ± .0035	10

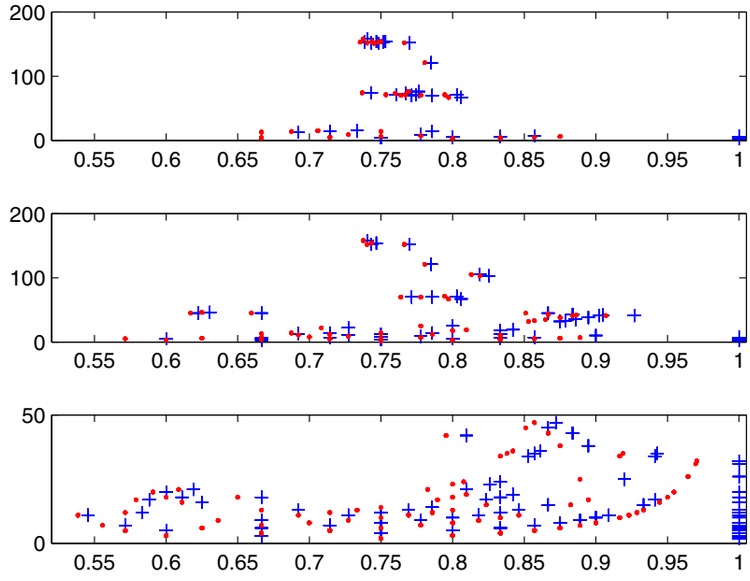


Figure 4.3: The purity of the leaves against their size for the spect data set and decision trees no. 1, 5, and 10 (from top to bottom). Pluses represent original data points, while the dots represent the Laplace corrected data points.

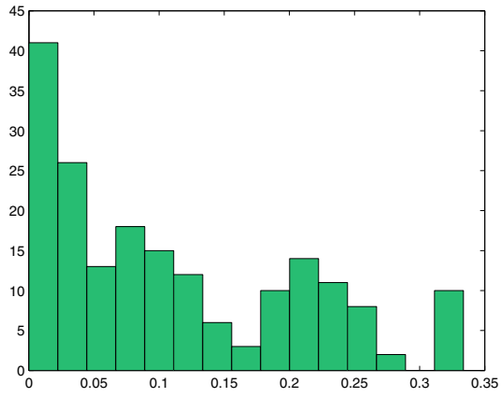


Figure 4.4: Histogram of the average change in rank position (as a fraction) for instances with extreme scores. Results are shown for all ten decision trees on the twenty data sets, so the sum of the bar heights is 200.

Table 4.3: The average number of distinct scores and standard deviations of decision trees no. 1, 5, and 10. Results are shown in two parts: without and with Laplace correction. The fourth column of each part lists the decision trees that produce the most distinct scores.

Data set	Without Laplace correction				With Laplace correction			
	1	5	10	max	1	5	10	max
1	46.3 ± 3.1	168.5 ± 6.9	989.7 ± 6.0	10	50.4 ± 2.9	182.5 ± 7.0	1251.8 ± 9.5	10
2	3.1 ± 0.3	4.2 ± 0.2	10.7 ± 0.4	10	3.1 ± 0.3	4.2 ± 0.2	15.3 ± 0.4	10
3	5.4 ± 0.3	6.4 ± 0.3	7.2 ± 0.3	7	5.6 ± 0.4	7.3 ± 0.4	11.3 ± 0.2	10
4	1.0 ± 0.0	1.5 ± 0.2	42.8 ± 1.2	10	1.0 ± 0.0	1.9 ± 0.4	78.2 ± 1.4	10
5	5.3 ± 0.4	9.2 ± 0.5	16.9 ± 0.5	10	5.4 ± 0.4	9.3 ± 0.5	25.5 ± 1.3	10
6	6.8 ± 1.2	20.2 ± 0.3	17.6 ± 0.5	9	6.7 ± 1.1	23.7 ± 0.3	27.9 ± 0.7	10
7	7.1 ± 0.3	8.5 ± 0.2	7.5 ± 0.3	5	7.7 ± 0.4	9.4 ± 0.3	12.7 ± 0.3	10
8	3.9 ± 0.1	4.2 ± 0.2	25.9 ± 0.9	10	3.9 ± 0.1	4.3 ± 0.2	28.0 ± 0.8	10
9	4.5 ± 0.3	5.5 ± 0.2	8.8 ± 0.3	10	4.5 ± 0.3	5.5 ± 0.2	9.1 ± 0.3	10
10	4.0 ± 0.1	4.3 ± 0.2	4.4 ± 0.2	8	6.2 ± 0.3	7.2 ± 0.3	7.6 ± 0.3	10
11	7.5 ± 0.2	6.3 ± 0.2	4.5 ± 0.3	1	20.0 ± 0.3	22.4 ± 0.5	24.8 ± 0.5	10
12	6.6 ± 1.1	11.5 ± 0.6	11.2 ± 0.5	3	6.6 ± 1.1	16.6 ± 0.4	17.3 ± 0.4	10
13	3.9 ± 0.3	3.0 ± 0.2	4.7 ± 0.4	10	12.2 ± 0.6	15.1 ± 0.5	20.6 ± 0.6	10
14	1.0 ± 0.0	4.6 ± 1.4	13.7 ± 0.5	8	1.0 ± 0.0	6.8 ± 2.2	21.2 ± 0.4	9
15	9.3 ± 0.6	11.1 ± 0.6	11.6 ± 0.4	8	10.4 ± 0.9	13.1 ± 0.9	14.8 ± 0.8	10
16	10.9 ± 0.7	15.1 ± 0.6	16.8 ± 0.5	8	14.2 ± 1.4	22.6 ± 1.2	28.3 ± 0.6	10
17	4.5 ± 0.1	4.3 ± 0.1	4.2 ± 0.1	2	7.9 ± 0.3	8.0 ± 0.3	8.1 ± 0.3	10
18	20.1 ± 0.5	26.6 ± 0.6	27.5 ± 0.4	7	31.7 ± 0.6	45.8 ± 0.9	53.1 ± 0.7	10
19	3.4 ± 0.2	6.1 ± 0.5	9.5 ± 0.3	10	3.4 ± 0.2	6.3 ± 0.5	13.4 ± 0.6	10
20	13.9 ± 0.6	14.4 ± 0.4	11.1 ± 0.3	7	25.5 ± 1.4	33.2 ± 0.5	32.9 ± 0.5	9

Now, consider that a leaf L_i is further split into two leaves L_{i1} and L_{i2}. Without loss of generality, we assume that $d_{i1} = p_{i1}/n_{i1} \geq p_{i2}/n_{i2} = d_{i2}$, where p_{i1} is the number of positive examples in L_{i1} and the other quantities are defined analogously. In the original ROC curve, the segment S_i is replaced by two segments S_{i1} and S_{i2} such that S_{i1} shares the starting point and S_{i2} the end point of S_i. Obviously, the curve thus obtained dominates the original curve, since S_i runs below $S_{i1}|S_{i2}$ while the rest of the curve remains unchanged. Therefore, the area under the modified curve is larger than the area under the original curve (or at least equal if $d_{i1} = d_{i2}$). The new area, however, may not correspond to the AUC of the modified tree since the segments $S_1, \ldots, S_{i-1}, S_{i1}, S_{i2}, S_{i+1}, \ldots, S_m$ are not necessarily well-ordered, which means that at least one of the conditions $d_{i-1} \geq d_{i1}$ and $d_{i2} \geq d_{i+1}$ might be violated. Hence, the question arises whether a locally beneficial modification has also a positive global effect on the AUC.

Lemma 4.1. *Consider a piecewise linear continuous curve consisting of m segments $S_1, \ldots, S_m$ and let $d_j \geq 0$ be the slope of S_j. If $d_{j-1} < d_j$, then swapping segments S_{j-1} and S_j can only increase the area under the curve.*

Proof. Since the rest of the curve remains unaffected, it suffices to consider the change of the area of the intersection between the area under the original curve and the rectangle defined by the diagonal that starts in the starting point of S_{j-1} and ends in the endpoint of S_j. Thus, subsequent to a suitable transformation, we can assume without loss of generality that S_{j-1} starts in $(0, 0)$ and S_j ends in $(1, 1)$. With (a, b) being the point where the two segments meet, we must have $b \leq a$ since the slope of S_{j-1} is smaller than the slope of S_j, which means that (a, b) must be located below the diagonal. After swapping the two segments, they will meet in the point $(1 - a, 1 - b)$, which is then located above the diagonal. Thus, $S_j|S_{j-1}$ runs above $S_{j-1}|S_j$ and, therefore, the area under the curve can only increase. □

Theorem 4.2. *Splitting a leaf can only increase the empirical area under the ROC curve of a decision tree.*

Proof. We have seen that splitting a leaf L_i and replacing the corresponding segment S_i by S_{i1} and S_{i2} (in the correct order) leads to a curve that runs above the original ROC curve and, therefore, covers a larger area. The new AUC is given by the area under the curve that is obtained after re-ordering the segments in decreasing order according to their slopes. This re-ordering can be achieved by repeatedly swapping the segment S_{i1} with its left neighbour and, likewise, repeatedly swapping the segment S_{i2} with its right neighbour. The previous lemma has shown that, in each of these steps, the area under the curve can only increase. Thus, the theorem immediately follows by induction over the number of swaps that are needed to bring the segments into a proper ordering. □

As shown by the previous result, a local improvement of a single segment is also globally beneficial in terms of AUC. Restrictively, however, one has to consider that our result refers to the *empirical* AUC and not the true AUC. In fact, the score associated with a leaf as well as the slope of the corresponding segment are only

estimations that may deviate from the real values. In the true ROC curve of a decision tree, the length of a leaf's segment corresponds to the *probability* that an instance falls into that leaf, which is estimated by the relative frequency in the training set. Likewise, the true slope δ_i of the segment is determined by the conditional probabilities of positive and negative instances in the leaf, and therefore may deviate from the estimation d_i derived from relative frequencies. As a result, the ordering of the segments according to the d_i may not coincide with the ordering according to the δ_i. In this case, the true ROC curve is non-convex and, therefore, the ranking performance in terms of AUC suboptimal. Formally, a non-convexity is caused by an *inversion*, that is, a pair of leaves (L_i, L_j) such that $d_i > d_j$ but $\delta_i < \delta_j$.

Nonetheless, even by looking at the problem from this point of view, there are two other convincing explanations for our finding that, in general, having more scores is better than having less.

- First, the ROC curve that can be obtained by splitting a segment, and hereby increasing the number of scores, is at least potentially better than the original curve. Roughly speaking, the more small segments are available (i.e., the less ties among instances), the more convex the curve can become. To illustrate, consider the two extreme cases. When there is only a single leaf, then the ROC curve consists of a single segment, namely the diagonal, and the AUC is 0.5. On the other hand, when each instance is comprised by a single leaf, the AUC can become arbitrarily close to 1, given that the leaves are correctly ordered.

- Second, the main pitfall preventing from a high AUC in the case of many leaves is inversions due to wrongly estimated scores, which may turn potential convexities into actual concavities. However, given the assumption that the estimation errors are bounded, inversions are likely to have a more local effect, i.e., they concern only leaves that are "neighboured" in terms of their scores. Again, this means that having more scores is better, as it increases the probability that two leaves are well-ordered (there is a smaller probability that two randomly chosen leaves are neighboured and, therefore, potentially inverted).

To illustrate and further investigate these two points, we conduct a simple but powerful experiment on synthetic data in the next subsection.

4.5.2 Simulation Studies with Synthetic Data

We generate a random binary list of 1000 elements where a 1 (0) indicates that the corresponding instance is positive (negative). We assume that these instances belong to the same leaf and start applying an iterative procedure that takes a random "leaf" and splits it in two. A fixed success parameter $u \in [0.5, 1]$ is used to arrange the instances in the new leaves such that the average fraction of positives in one leaf is u and the average fraction of negatives in the other leaf is also u. At each iteration we compute the true and empirical AUC as follows. By definition, the true AUC is determined by the frequency of positives in all leaves. The empirical AUC is determined by adding noise to each of these frequencies. Noise is randomly

drawn from an interval $[-h, +h]$, so if a true frequency is a, then its estimation lies in $[a-h, a+h]$. We stop the leaf splitting iteration when there is no leaf left comprising more than two instances. The complete process is repeated for twenty times and we report average test statistic values. We already note that the results do not strongly depend on the size of the binary list or the class distribution.

To illustrate and support our formal conjecture, in Fig. 4.5 we show the average empirical AUC for a fixed $h = 0.2$ and $u = 0.5, 0.6, 0.7$, respectively, as a function of the number of leaves. Clearly, splitting leaves (i.e., breaking ties or splitting segments) leads to higher AUC, and the better the tie breaking, the larger the gains in AUC. Corresponding standard deviations are extremely small, decreasing with the number of leaves, and for clarity have been omitted in the figure.

The influence of the estimation errors is illustrated in Fig. 4.6. Here, we have fixed $u = 0.2$ and we show the average difference between true AUC and empirical AUC for $h = 0.1, 0.2, 0.4, 1$, respectively (with $h = 1$ implying completely random scores). Clearly, even for quite high values of h the difference between true and empirical AUC is small, and after an early point decreasing when the number of leaves increases (of course, except for completely random scores). In other words, leaf-splitting seems to be tolerant toward estimation errors of probabilities for the positive class in the sense that only the *ordering* of scores is important. Indeed, as long as an estimation error does not cause an inversion of scores with respect to the true probabilities, there is no negative effect on the AUC.

4.5.3 Conclusions from the Theoretical Results

From our theoretical analysis we may conclude that the empirical AUC increases with the number of scores produced by a tree, at least when these scores are better than random approximations of the true conditional probabilities in the leaves. This result explains why unpruned trees or only slightly pruned trees in combination with Laplace correction in the leaves have been shown to be very good rankers. It also explains why other methods such as bagging and propagating instances through various branches of a decision tree leads in general to an improvement in AUC. The experiments with synthetic data support our conjecture that local tie breaking is beneficial for the AUC, and that it is robust toward estimation errors since we are only interested in a correct ordering of the scores. In the next section, we provide even stronger evidence for the usefulness of the formal analysis by means of presenting a simple new method to AUC-optimising decision trees.

4.6 Score Perturbation for AUC-Optimising Trees

To continue our investigation on the conjecture that the number of distinct scores produced by a decision tree is essential for its AUC, we propose and evaluate in this section a simple method for AUC-optimising decision trees. The method averages over the score of a number of copies of the test instance, where each copy is corrupted by random noise. In this way, the number of distinct scores significantly increases.

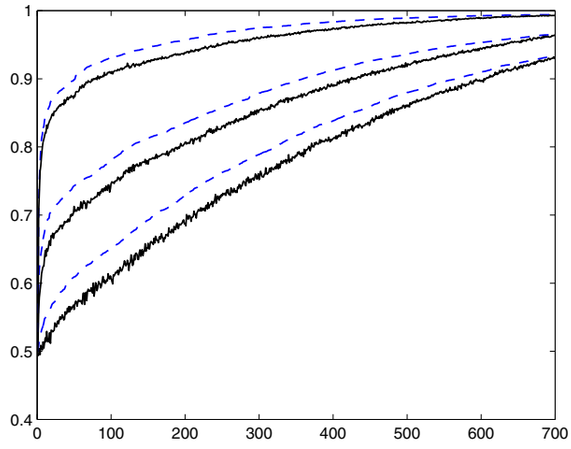


Figure 4.5: Results of the first simulation study: true AUC (dashed) and empirical AUC (solid) where curves more to the point (0, 1) correspond to higher values of u.

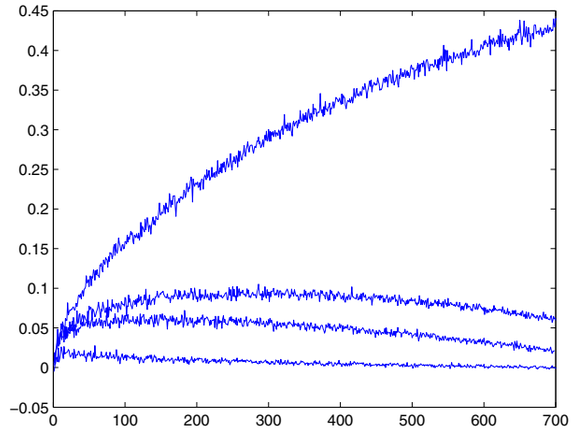


Figure 4.6: Results of the second simulation study: difference between true AUC and empirical AUC where higher curves correspond to higher values of h.

For simplicity, we assume that instances consist of only numerical features. Given a test instance $\boldsymbol{x} \in \mathcal{X}$, we perturb its feature vector by adding noise. The resulting perturbed instance is denoted by $\boldsymbol{x}_{\boldsymbol{\varepsilon}} = \boldsymbol{x} + \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_d)$ a random vector of noise values for each feature. Components ε_i are drawn independently from Gaussian distributions $N(0, \lambda_i)$. It is natural to set the parameter $\lambda_i \geq 0$ as a function of the standard deviation of the i-th feature. The score assigned to the perturbed instance $\boldsymbol{x}_{\boldsymbol{\varepsilon}}$ is recorded and the process is repeated for T times. The final score assigned to the test instance $\boldsymbol{x}$ is the mean of the scores of its perturbed versions.

Figure 4.2(c) shows the AUC values obtained by the perturbation method for four representative data sets. Table 4.4 summarises the results for all data sets that contain at least one numerical feature by reporting the average improvements in AUC and its standard deviation. The number in the fourth column is in bold when the perturbation method produces a decision tree that is smaller and has higher AUC than the best decision tree obtained in our first set of experiments. The experimental setup is kept identical and we use $T = 200$ perturbations. We roughly approximated the optimal value for λ_i as follows. First, the training data is standardised to zero mean and unit variance for each feature (test instances are transformed accordingly). Then, we tried a single noise parameter for all features, namely $\lambda_i = \lambda \in \{0.0001, 0.00015, 0.001, 0.005, 0.01, 0.1, 0.15, 0.2\}$. The reported results correspond to the best λ as measured on a validation set (20% of the training set). From the results, we make three observations.

- First, comparing the results with and without Laplace correction, we clearly have that decision trees without Laplace correction benefit most from the perturbation method. This was to be expected since these trees by themselves are not able to produce many distinct scores. Nonetheless, in general, it remains best to use Laplace correction in the leaves in order to distinguish between small and large leaves. More specifically, the null hypothesis that Laplace correction does not change the ranking performance of the perturbation method is rejected by the Wilcoxon signed-ranks at the 5% significance level.

- Second, considering the sign of the test statistics, the perturbation method leads to higher AUC values along all pruning levels when compared to the standard trees (independent whether Laplace correction is used). This result holds at the 1% significance level and hence is strong proof for our conjecture that a reasonable local tie breaking renders trees that are very good rankers.

- Third, since many pruning levels are printed in bold, we can almost always use the perturbation method with a tree that is much smaller and a better ranker than the best (often unpruned) tree in our first set of experiments. For example, the tree #7 in combination with the perturbation method is a better ranker than a standard unpruned tree (#10) at the 5% significance level.

As a final remark, we note that the number of different scores assigned by the perturbation method is many times higher than a conventional decision tree can produce (even when the latter uses Laplace correction; compare Figs. 4.2(b) and

4.2(d), and see Table 4.5 for details). Thus, there is a clear positive correlation between number of distinct scores assigned to instances and the AUC, as was to be expected from the formal analysis and all our experiments so far. By exploiting this correlation we have experimented with a simple but powerful method to increase the ranking performance of decision trees.

4.7 Generalisation to Other Classifiers

Many machine learning classifiers have been shown to outperform decision trees in the standard classification setting (Caruana and Niculescu-Mizil, 2006). Interestingly, these classifiers often already produce many distinct scores, or at least, have also a parameter that allows to control the number of scores. Since our formal analysis of optimising AUC is independent of the learning algorithm, it is interesting to investigate to what extent the AUC of these other classifiers can be improved.

As an example, consider a k-nearest neighbour classifier ($k \geq 1$) where predictions of nearest neighbours are weighted by the inverse of their distance to the test instance. According to our expectation, we found that this distance-weighted version is many times better in terms of AUC than the conventional k-NN classifier since the latter resulted in much more ties among ranked instances. Also, the number of distinct scores of the distance-weighted classifier often stabilized for small values of k already, and correspondingly, the AUC did not change anymore. These results are in perfect agreement with our formal analysis and experiments with decision trees.

We conjecture that the distance-weighted k-NN is a better ranker than a decision tree because it produces more accurate scores (Caruana and Niculescu-Mizil, 2006) and more distinct scores (often no ties at all). To verify this conjecture, we experimented with the classifier in the same spirit as we did for decision trees except that we do not vary the pruning level but the number of nearest neighbours $k = 1, \ldots, 10$. Without going into much detail about the experimental results, we mention the following two observations.

- First, compared to the best results of our decision tree experiments (excluding the perturbation method), the number of scores is much higher for k-NN. Also, this classifier has in general highest AUC. We count 14 wins and 6 losses and the average increase (decrease) in value for a win (loss) is .06821 (.02524). This result is significant at the 5% significance level. In terms of accuracy, we have only 9 wins and 11 losses and the average increase (decrease) for a win (loss) is .06057 (.02185). Optimising the AUC is thus not always the same as optimising accuracy, which is even more compelling for decision trees.

- Second, compared to the perturbation method, weighted k-NN produces a larger number of distinct scores. Also, we count for both AUC and accuracy 7 wins and 5 losses. The average win (loss) is around .04231 (.02167) for both performance metrics; not significant. Of course, it is possible to improve the perturbation method by tuning the noise level λ_i for each feature separately.

Table 4.4: Improvement in AUC and its standard deviation (values are separated by a /) gained by the perturbation method for decision trees no. 1, 5, and 10. Results are shown in two parts: without and with Laplace correction. The fourth column of each part lists the decision trees that obtain the highest AUC. A number is in bold when the perturbation method results in a decision tree that is smaller and has higher AUC than the best decision tree so far.

Data set	Without Laplace correction				With Laplace correction			
	1	5	10	max	1	5	10	max
1	.0022 / .0050	.0050 / .0045	.0203 / .0074	**7**	.0022 / .0058	.0039 / .0050	.0011 / .0033	**9**
3	.0207 / .0132	.0193 / .0200	.0269 / .0357	**10**	.0143 / .0139	.0096 / .0130	.0069 / .0137	**10**
5	.0051 / .0200	.0096 / .0229	.0470 / .0455	**9**	.0053 / .0211	.0062 / .0200	.0115 / .0113	**9**
6	.0004 / .0913	.0316 / .0508	.0393 / .0514	2	.0004 / .0919	.0082 / .0298	.0034 / .0400	**4**
7	-.0017 / .0658	.0231 / .0628	.0325 / .0754	**8**	-.0025 / .0786	.0028 / .0667	.0025 / .0921	**10**
8	.0000 / .0675	.0000 / .0675	.0003 / .0587	1-5	.0000 / .0675	.0000 / .0675	.0006 / .0557	10
10	.0338 / .0618	.0412 / .0525	.0319 / .0502	**10**	.0167 / .0461	.0111 / .0477	.0102 / .0506	**10**
12	.0563 / .0497	.0461 / .0554	.0401 / .0526	**10**	.0595 / .0463	.0453 / .0442	.0342 / .0477	**6**
15	.0496 / .0529	.0556 / .0283	.0543 / .0370	**4**	.0440 / .0497	.0373 / .0341	.0323 / .0371	**10**
16	.0139 / .0322	.0211 / .0271	.0356 / .0363	**10**	.0039 / .0360	.0080 / .0264	.0043 / .0116	10
17	.1046 / .0733	.0905 / .0843	.0946 / .0799	1	.0537 / .0561	.0555 / .0501	.0537 / .0547	**10**
18	.0260 / .0102	.0266 / .0096	.0278 / .0104	**2**	.0162 / .0094	.0029 / .0042	-.0013 / .0051	**7**

Table 4.5: Relative increase in the number of distinct scores gained by the perturbation method for decision trees no. 1, 5, and 10 (e.g, a value of 10 means that the perturbation method yields ten times as many distinct scores than the conventional decision tree). Results are shown in two parts: without and with Laplace correction. The fourth column of each part lists the decision trees that obtain the highest number of distinct scores.

Data set	Without Laplace correction				With Laplace correction			
	1	5	10	max	1	5	10	max
1	15.2 ± 4.5	8.6 ± 0.9	3.5 ± 1.1	10	14.0 ± 4.8	7.9 ± 0.9	2.7 ± 0.6	10
3	3.8 ± 0.3	3.6 ± 0.5	3.9 ± 0.6	10	3.7 ± 0.3	3.2 ± 0.4	2.5 ± 0.7	10
5	2.0 ± 0.7	2.2 ± 1.1	2.2 ± 0.8	10	1.9 ± 0.6	2.2 ± 1.1	1.4 ± 0.3	10
6	1.4 ± 0.3	1.5 ± 0.8	2.0 ± 0.6	10	1.4 ± 0.3	1.3 ± 0.9	1.3 ± 0.3	10
7	1.7 ± 0.6	1.8 ± 0.6	2.2 ± 0.4	10	1.5 ± 0.5	1.6 ± 0.4	1.3 ± 0.4	10
8	1.0 ± 0.2	1.0 ± 0.2	1.0 ± 0.2	10	1.0 ± 0.2	1.0 ± 0.2	0.9 ± 0.2	10
10	5.3 ± 1.8	6.0 ± 1.3	5.9 ± 1.3	8	3.4 ± 1.1	3.6 ± 0.5	3.4 ± 0.5	10
12	3.8 ± 0.2	2.6 ± 0.3	2.7 ± 0.3	6	3.8 ± 0.2	1.8 ± 0.3	1.7 ± 0.2	6
15	5.5 ± 1.2	5.5 ± 0.9	5.7 ± 0.5	10	5.0 ± 0.7	4.7 ± 0.6	4.5 ± 0.3	10
16	5.0 ± 1.0	5.1 ± 1.0	5.0 ± 0.9	10	3.8 ± 0.6	3.4 ± 0.7	3.0 ± 0.7	10
17	3.9 ± 0.6	4.0 ± 0.7	4.1 ± 0.8	2	2.2 ± 0.2	2.2 ± 0.2	2.1 ± 0.2	5
18	17.0 ± 1.5	13.9 ± 1.1	13.8 ± 1.5	10	10.8 ± 1.0	8.1 ± 0.7	7.1 ± 0.8	10

These findings are consistent with some empirical results presented by Caruana and Niculescu-Mizil (2004), Jiang, Zhang, and Su (2005) and Huang and Ling (2005). Moreover, we offer an explanation for the results. We believe that the ranking success of other classifiers such as boosted decision trees and neural nets can be explained analogously to nearest neighbour as we just did. The main point is that the more scores are produced and the more accurate these scores are, the higher the AUC.

4.8 Chapter Conclusions

In this chapter, we focused on RQ 2: *How can the AUC of an interpretable and comprehensible classifier, such as decision trees, be optimised?* We have answered this question by means of an extensive set of experiments and a formal analysis. From the results obtained, we may conclude that the ranking performance of decision trees can be improved by increasing the number of scores that they produce. This measure works as long as the scores are better-than-random estimations of the true conditional probabilities in the leaves. The robustness toward estimation errors was tested empirically and found to be more than reasonable. In fact, simply adding noise to features already improves drastically the AUC of decision trees.

Our answer to the second research question provides a unifying framework in which the success of other methods to learn AUC-optimising decision trees can be explained. We mention two important contributions from this point of view. First, in contrast to previous conjectures, the advantage of Laplace correction is not (necessarily) a better probability estimation, but a reasonable local tie breaking effect that comes along with an increased number of scores. This also explains why generalisations of the Laplace correction, such as the m-estimate, essentially yield the same results (no further improvement is obtained). Second, decreasing the level of pruning increases the number of segments of the ROC curve, and thereby leads to higher AUC provided that the empirical scores are sufficiently good estimations of the true slopes of the segments (i.e., the true class probabilities). In fact, the relationship between AUC and pruning level is not necessarily monotone, as it was previously assumed to be, especially when Laplace correction is turned off.

The findings in this chapter are not restricted to decision trees. We have chosen decision tree learning since its bad ranking performance has received a lot of attention in the literature. However, the formal analysis is classifier independent, and thus, generalises to other methods and classifiers in a straightforward way. We have illustrated this with a distance-weighted k-nearest neighbour classifier.

Chapter 5

The ROC Isometrics Approach

In some domains, it may happen that the expected classification performance of a classifier is not sufficient for implementation in real-life practice. In other words, too many incorrect classifications are made. Boosting and related ensemble methods are often used to increase performance, but as a consequence, the resulting classifier is not easily interpretable and may have become much slower. In this chapter, we are interested in developing an efficient approach that adjusts an existing classifier such that it is able to guarantee the performance as defined *a priori* by a domain expert. Our result provides an answer to the third research question.[1]

5.1 Presetting Classification Performance

There is a higher than normal risk associated with applying a classifier that maps instances to class labels in domains characterised by high error costs. Therefore, in the previous two chapters, we elaborated on ranking instances from most likely positive to most likely negative. Such a ranking allows among others to focus on the most interesting cases, which is clearly useful in a field such as law enforcement.

However, despite the benefits of the ranking setting, it is still unknown *a priori* how many and which mistakes the classifier made among the selected instances. Only after a domain expert investigated each case separately, it is known which cases were indeed of interest. Even more importantly, we stress that the number of allowed mistakes is assumed to be very low and the task to be difficult. This may imply that conventional classifiers do not have sufficient classification performance for an actual implementation in real-life practice.

Therefore, a different point of view to the problem statement is not to preset the number of cases to be investigated, but to preset the classification performance. So, discrete classifiers are applied only if they are able to guarantee a preset classification

[1] This chapter is based on an article by Vanderlooy, Sprinkhuizen-Kuyper, Smirnov, and van den Herik (2009). The article uses and extends earlier work by Vanderlooy, Sprinkhuizen-Kuyper, and Smirnov (2006b; 2006c).

performance on each class. The values of these performances are chosen in such a way that the costs of incorrect classifications that still may occur are acceptable. Clearly, of special interest to us are the cases in which the preset performance is higher than the expected performance of the classifier. Boosting and other ensemble techniques are often used to increase performance, but the resulting classifier is not easily interpretable and has become (much) slower. Moreover, in practice, there is not a guarantee that such techniques can eventually reach the desired performance level (Bickel, Ritov, and Zakai, 2006).

Therefore, in this chapter, we are interested in answering our third research question (RQ 3): *Can we develop a feasible approach by which a classifier is constructed that guarantees a preset classification performance on each class?* An affirmative answer to this question implies that classifiers can become reliable (we know what to expect from them) and they can thus be safely applied in practice. The presented research will assume the binary classification setting, and henceforth, when we talk about performance, we always mean classification performance.

Our answer to the research question is decomposed into two stages, each stage leading to one or more contributions. First, we analyse the effect on performance when instances with uncertainty in the true class label are left unclassified. The motivation of this stage is that a classifier benefits from being self-aware in the sense that it knows what it has learned so far and what not. This means that the classifier is able to distinguish between instances that are learned with sufficient accuracy and those of which the class label is still unknown. By filtering out uncertain instances and leaving them unclassified (this is called *abstention*) we often boost performance significantly. To obtain a better understanding of this procedure, we visualise the effect of abstention on the classification performance. This is done by transforming an ROC curve into a new curve that represents the performance statistics when abstention is applied. We analyse when and where this new ROC curve is better than the original one, resulting in several new results and guidelines for practice.

In the second stage, we wish to fine-tune the effect of abstention to identify those instances that should be left unclassified in order to obtain a reliable classifier. A trial-and-error procedure is clearly not feasible. Therefore, we propose an approach to find the minimum set of instances that should be left unclassified such that the resulting classifier guarantees (up to statistical fluctuations) a preset performance. Our approach to construct reliable classifiers, called the *ROC isometrics approach*, uses the classifier information provided by an ROC curve in combination with so-called isometrics. Our theoretical results are verified by an extensive empirical evaluation of the approach using benchmark data sets.

The remainder of the chapter is organised as follows. In Section 5.2 we review some concepts that have not been introduced so far. A formal analysis of abstention is given in Section 5.3. In Section 5.4 we introduce and show formally the effectiveness of the ROC isometrics approach. An experimental verification of the approach is provided in Section 5.5 and in Section 5.6 we compare it with related work. Finally, Section 5.7 concludes the chapter and provides an answer to the research question.

We note that, for ease of readability, the proofs of all theorems are excluded from the main text and gathered in Appendix A.

5.2 Abstention and Performance Evaluation

In this section we review three concepts used throughout the chapter that have not been introduced so far, namely: abstaining classifiers (5.2.1), skew-sensitive performance evaluation (5.2.2), and ROC isometrics (5.2.3).

5.2.1 Abstaining Classifiers

In complex real-life applications, a classifier often encounters instances to be classified that are different from instances encountered during the training phase. Especially when incorrect classifications have high costs, it becomes desirable to abstain from classifying instances for which there is uncertainty in the true class label (in other words, the cost of abstention becomes smaller than the cost of incorrect classifications). Henceforth, we refer to these instances as *uncertain instances*.

An *abstaining classifier* is a classifier that can abstain from classifying uncertain instances (we refer for references to the section about related work). Such a classifier can improve the classification performance significantly, even for low abstention rates. It can be considered as simulating the behaviour of human experts. For example, in medical diagnosis, an expert does not state a possibly incorrect diagnosis but she says "I do not know" and possibly performs more tests. Similarly, uncertain instances that are left unclassified by an automated system can be discarded from the system, passed to a human for classification, or classified by another classifier (Muzzolini, Yang, and Pierson, 1998; Frélicot and Mascarilla, 2002; Ferri, Flach, and Hernández-Orallo, 2004).

For an abstaining classifier to be successful, it has to distinguish between instances that are learned with sufficient accuracy and those of which the class labels are still unknown. Hence, the classifier should have some form of *self-awareness* in the sense that it should know what it has learned so far. We represent this knowledge by the score of an instance. Thus, uncertain instances are considered as instances with scores that do not clearly indicate the correct classification. This allows to implement an abstaining classifier as a reject rule with two thresholds $a > b$. Instances with score at least a are considered to be evidently positives, while evidently negatives are those with score at most b. All remaining instances are left unclassified. We note that $a = b$ results in a conventional discrete classifier. Henceforth, we denote the number of unclassified positive instances by UP and the number of unclassified negative instances by UN. The proportion of abstention is defined by the *unclassified positive rate* (upr) and the *unclassified negative rate* (unr):

$$\begin{aligned} upr &= \frac{UP}{TP + FN + UP} \\ unr &= \frac{UN}{FP + TN + UN} \end{aligned} \quad . \tag{5.1}$$

Figure 5.1 shows an example of the effect of an abstaining classifier on the classification performance. Clearly, instance classifications have become correct because

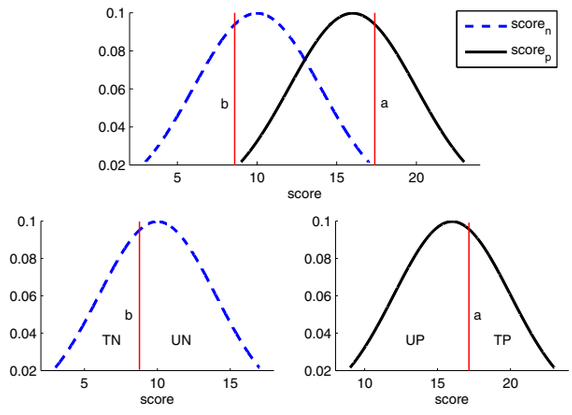


Figure 5.1: Visualising abstention with the score density functions of the negative class and the positive class: (above) two possible thresholds are indicated by vertical lines, and (below): the relation between the thresholds and the performance statistics is given by the four distinct areas.

instances in the overlap of the score density functions are left unclassified. Decreasing the number of unclassified instances is achieved by (1) decreasing the value of a, (2) increasing the value of b, or (3) both. Decreasing the value of a results in fewer unclassified positive instances but also in more false positives. Increasing the value of b results in fewer unclassified negative instances, although also in more false negatives. What to prefer depends on the requirements of the application domain. We note that these requirements can change over time; see again Section 1.3.

The goal of abstaining classifiers is to keep the total number of unclassified instances as low as possible while still guaranteeing a significantly higher performance than that obtained by a conventional classifier without reject rule. In this chapter, we are interested in formalising and benefiting from the relationship between the thresholds a and b, the resulting abstention rate, and the gain in performance.

5.2.2 Skew-sensitive Performance Evaluation

A central role in the performance evaluation of a classifier is played by the *cost distribution* and the *class distribution* of the application domain (Ting, 2002; Weiss and Provost, 2003; Fawcett and Flach, 2005). Most of the widely-used performance metrics do not consider these two distributions. Therefore, under some conditions, many metrics may be (or often are) considered as inappropriate to measure classification performance. For instance, it is well-known that evaluating classifiers by means of accuracy has two severe shortcomings (Provost *et al.*, 1998). First, accuracy assumes that the class distribution is constant and relatively balanced. Optimising classifiers in terms of accuracy when the class distribution is (extremely) skewed

often prefers trivial classifiers that ignore the minority class. Large class skews are however common and the minority class is often the class of interest. Second, accuracy assumes that the costs of a false negative and a false positive are equal. This assumption is unrealistic since in practice a false positive is often more expensive (or less expensive, depending on the definition of the class) than a false negative. We note that costs can even vary among instances of the same class (Fawcett, 2006).

Clearly, the cost and class distribution should be taken into account when evaluating classifiers. For this reason, the *skew ratio* is defined to express the relative importance of the negative class versus the positive class (Elkan, 2001; Flach, 2003):

$$c = \frac{c(p,n)}{c(n,p)} \frac{\pi_n}{\pi_p} \quad , \tag{5.2}$$

where $c(p,n)$ and $c(n,p)$ are the costs of a false positive and a false negative, respectively. The probabilities of a negative instance and positive instance are denoted by $\pi_n = N/(P+N)$ and $\pi_p = P/(P+N)$. The class distribution is $\pi_n/\pi_p = N/P$. We may assume that the cost distribution is given only by the incorrect classification costs since benefits of correct classifications can be incorporated by substracting them from the corresponding errors (Friedel, Rückert, and Kramer, 2006).

The skew ratio can be used as a user-defined parameter in performance metrics. This allows us to measure classification performance in any possible scenario of cost distributions and class distributions. For instance, we may set the skew ratio equal to the class distribution in the training set if we believe that this is representative for the application domain. If we believe that this is not the case, then we may insert the correct class distribution into the skew ratio. The same holds for the cost distribution. In general, if $c < 1$ then the positive class is most important, and if $c > 1$ then the negative class is most important. In the following, we set c equal to the class distribution to keep equations and derivations easy to comprehend. Our results are also valid when a different class or cost distribution is incorporated.

5.2.3 ROC Isometrics

The ROC isometrics are defined as curves in ROC space that connect points with the same value for a (skew-sensitive) performance metric. So, they can be considered as geometrical representations of the performance metrics.

Isometrics have been introduced by Vilalta and Oblinger (2000) in order to understand the bias of performance metrics. However, their study was restricted to information gain and no connection to ROC analysis was made. Flach (2003) was the first to demonstrate this connection for a variety of metrics and Fürnkranz and Flach (2005) used isometrics for understanding rule-learning algorithms. We note that the main focus of these works lies in analysing performance metrics.

Our exposition of isometrics, to be given later on, will largely follow the latter two works, except that we define two types of isometrics for each performance metric. In addition, our goal of using ROC isometrics is different since we will use them as a point of departure for defining reliable classifiers.

5.3 Effect of Abstention in ROC Space

In this section, we provide our results on how abstention can be visualised and analysed in ROC space. We describe the effect on the ROC curve of a classifier when it is transformed into an abstaining classifier (5.3.1). Then, we provide several conditions for dominance relations between the original curve and that of abstention (5.3.2).

5.3.1 Abstention ROC Curves

In the following, we consider a scoring classifier and its ROC curve. Henceforth, we denote this curve by the *original curve*. We recall that each point on the original curve is the result of applying some numerical threshold (or a range of thresholds) on the scores of instances. Without restriction we assume a convex curve; here we recall the theorem in Subsection 2.2.3. To transform the scoring classifier into an abstaining classifier, we have to define two thresholds a and b. These thresholds can be used to construct the ROC curve of the abstaining classifier by using the classified instances only. We call this curve the *abstention curve*.

We write (fpr_a, tpr_a) and (fpr_b, tpr_b) to denote the points on the original curve that correspond to the thresholds a and b of the abstaining classifier. An illustration is given in Fig. 5.2. Intuitively, the abstention curve is obtained by only considering thresholds for which the false positive rate is at most fpr_a or at least fpr_b. This corresponds with not "covering" the part of the curve between points (fpr_a, tpr_a) and (fpr_b, tpr_b) since, roughly speaking, the construction of this part involved instances that are now left unclassified. For this reason, we define the *uncovered part* as the part on the original curve between (fpr_a, tpr_a) and (fpr_b, tpr_b). The *covered part* is defined as the part from $(0,0)$ to (fpr_a, tpr_a) and from (fpr_b, tpr_b) to $(1,1)$. So, the covered part is the complement of the uncovered part. By definition, the unclassified positive rate and the unclassified negative rate are

$$\begin{aligned} upr &= tpr_b - tpr_a \\ unr &= fpr_b - fpr_a \end{aligned} . \tag{5.3}$$

The transformation from the original curve (conventional scoring classifier) to the abstention curve (abstaining classifier) is given in Theorem 5.1. The key idea is that a point on the covered part of the original curve is associated with a point on the abstention curve such that the corresponding discrete classifiers classify the same instances as positive or negative, depending on the location in the covered part. More specifically, when we are to the left of fpr_a, then a discrete classifier on the original curve is linked to a discrete classifier on the abstention curve in the sense that they classify the same instances as positive. A similar link exists between classifiers when located to the right of fpr_b on the original curve, except that the link is defined by classifying identical instances as negative. More details can be found in the proof of the theorem.

Theorem 5.1. *If the part between (fpr_a, tpr_a) and (fpr_b, tpr_b) of an ROC curve is not covered, and $0 < upr < 1$ and $0 < unr < 1$, then points (fpr_i, tpr_i) on this*

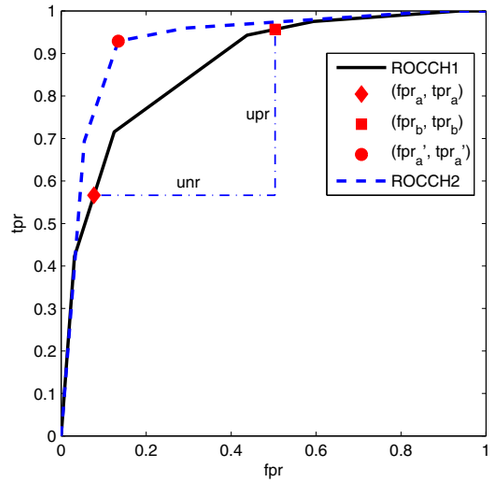


Figure 5.2: Visualising the effect of abstention in ROC space: the abstention curve (dashed curve) is obtained by not covering the part between points (fpr_a, tpr_a) and (fpr_b, tpr_b) of the original curve (solid curve). The length of the horizontal and vertical dash-dotted lines equals unr and upr, respectively.

original curve between $(0,0)$ *and* (fpr_a, tpr_a) *are transformed into points* (fpr'_i, tpr'_i) *on the abstention curve such that*

$$fpr'_i = \frac{fpr_i}{1-unr} \text{ , and } tpr'_i = \frac{tpr_i}{1-upr} \text{ .} \tag{5.4}$$

Also, points (fpr_i, tpr_i) *between* (fpr_b, tpr_b) *and* $(1,1)$ *are transformed into points* (fpr'_i, tpr'_i) *on the abstention curve such that*

$$fpr'_i = 1 - \frac{1-fpr_i}{1-unr} \text{ , and } tpr'_i = 1 - \frac{1-tpr_i}{1-upr} \text{ .} \tag{5.5}$$

We note that points (fpr_a, tpr_a) and (fpr_b, tpr_b) are transformed into the same point on the abstention curve. Figure 5.2 shows an example of a transformation with $upr = 0.34$ and $unr = 0.41$. The transformation of the endpoints of the uncovered part is also indicated. It should be clear that the effect of abstention on classification performance is specific to the data set and the classifier that is used. The original curves presented in this chapter will reflect a classifier with "moderate" performance allowing us to emphasize visually the effect of abstention. In general, we obviously have that the better the scores of a classifier are, the less need there is for a reject rule, i.e., the effect of abstention becomes smaller with original curves that approach the optimal point $(0,1)$.

Theorem 5.2. *If the original curve is convex, then the ROC curve of the abstaining classifier is also convex.*

Theorem 5.2 implies that we can restrict ourselves to convex ROC curves in the remainder of this chapter. Nonetheless, for ease of readability, we simply talk about curves instead of convex hulls. In the next subsection, we analyse the dominance relations between the original curve and the abstention curve.

5.3.2 Formal Analysis of Dominance Relations

Dominance relations among ROC curves indicate which curve delivers the best discrete classifiers for certain fixed *operating characteristics* (Fawcett, 2003). More specifically, applied to the current context, if the abstention curve dominates in a region of ROC space, then abstention yields the highest classification performance when applied in the corresponding scenario of cost distributions and class distributions (i.e., skew ratio). Clearly, we wish that the abstention curve dominates as much as possible since this implies that abstention gives better classification performance than was previously possible, and that it is robust to changes in the application domain (i.e., a change in skew ratio still implies better performance).

Theorem 5.3 shows that the abstention curve always dominates the original one on the part from $(0,0)$ to (fpr_a, tpr_a) if the unclassified negative rate is less than or equal to the unclassified positive rate. Theorem 5.4 provides a similar condition for dominance on the part from (fpr_b, tpr_b) to $(1,1)$. Both theorems conform with our intuition. For example, when we remove more and more of the positive instances by shifting a on the curve toward the left in ROC space, we expect to have increased the *tpr* for low *fpr* since the remaining positive instances are easy to classify.

Theorem 5.3. *If $unr \leq upr$ then the abstention curve dominates the original curve on the part from $(0,0)$ to (fpr_a, tpr_a).*

Theorem 5.4. *If $unr \geq upr$ then the abstention curve dominates the original curve on the part from (fpr_b, tpr_b) to $(1,1)$.*

Illustrations of these two theorems are given in Figs. 5.3(a) and 5.3(b), respectively. Since it is always true that $unr \leq upr$ or $unr \geq upr$, at least one of the two theorems is applicable. Thus, there is always a part where the abstention curve is dominating. Moreover, we have the following simple corollary.

Corollary 5.1. *If $unr = upr$ then the abstention curve dominates the original curve on the covered part.*

This corollary is important in practice, in particular when the skew ratio cannot be estimated precisely and/or when it is subject to strong change over time. The corollary guarantees that abstention leads always to the highest classification performance. At first sight, it may seem that applications possessing continuously evolving class and cost distributions are exceptional. However, on the contrary, such applications actually occur often (Ling and Sheng, 2010). The example of fraud detection is a nice illustration; see again Section 1.3 where we provided more information specific to law enforcement.

To complement these results, the next two theorems give conditions for dominance on the (complete) covered part when, respectively, the first line segment of the original curve is vertical and the last line segment is horizontal.

Theorem 5.5. *If the original curve contains a point* $(0, tpr_0)$ *with* $tpr_0 > 0$*, then the abstention curve dominates the original curve on the covered part if* $unr > upr$ *and* $tpr_a \leq \frac{1}{1-\frac{upr}{unr}} tpr_0$.

Theorem 5.6. *If the original curve contains a point* $(tnr_0, 1)$ *with* $tnr_0 = 1 - fpr_0 > 0$*, then the abstention curve dominates the original curve on the covered part if* $upr > unr$ *and* $1 - fpr_b \leq \frac{1}{1-\frac{unr}{upr}}(1 - fpr_0)$.

Figure 5.4(a) shows a situation where Theorem 5.6 applies. By combining Corollary 5.1 and Theorems 5.5 and 5.6, we can relax the condition for dominance on the covered part. A relaxation is desired since Corollary 5.1 does not often apply in practice. Our relaxation is given in the next corollary and is illustrated in Fig. 5.4(b).

Corollary 5.2. *If the original curve contains two points* $(0, tpr_0)$ *and* $(tnr_0, 1)$ *with* $tpr_0 > 0$ *and* $tnr_0 = 1 - fpr_0 > 0$*, and if* $\frac{fpr_0 - fpr_b}{1 - fpr_b} \leq \frac{unr}{upr} \leq \frac{tpr_a}{tpr_a - tpr_0}$*, then the abstention curve dominates the original curve on the covered part.*

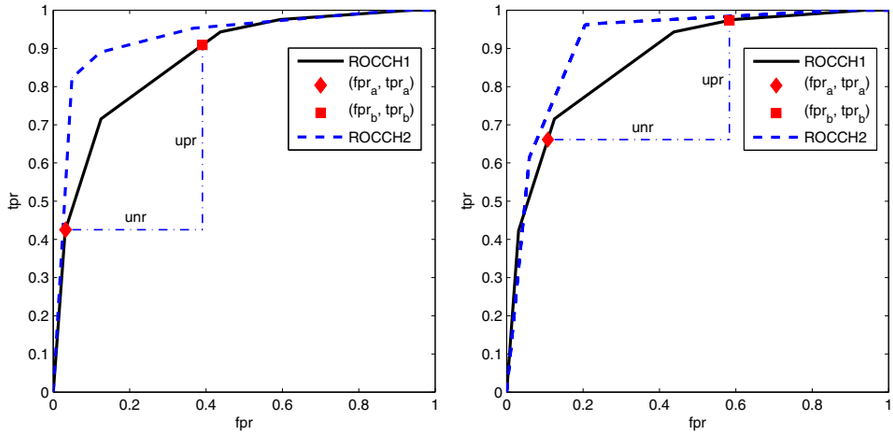
This corollary implies that the abstention curve dominates on the covered part when the first line segment is vertical, the last line segment is horizontal, and the unclassified positive rate is approximately the unclassified negative rate (how good this approximation should be depends on the lengths of the first and last line segment). We note that these conditions are often easier to satisfy in practice than the condition we had first in Corollary 5.1. Indeed, most scoring classifiers do not assign the highest score to a negative instance and the lowest score to a positive instance. Therefore, the corresponding curves have a first line segment that is vertical and a last line segment that is horizontal. It may be the case that the lengths of these two segments are small, but nonetheless Corollary 5.2 applies when we define an abstaining classifier such that $upr \approx unr$. This condition can be relaxed more and more for increasing values of tpr_0 and decreasing values of fpr_0.

5.4 How to Construct Reliable Classifiers

In this section, we introduce our approach to construct classifiers that guarantee a preset classification performance. We first discuss ROC isometrics (5.4.1), then we present our approach (5.4.2), and we provide a formal analysis (5.4.3). From the analysis we may conclude that this approach provides, at least in theory, an affirmative answer to the research question that we address in this chapter (5.4.4).

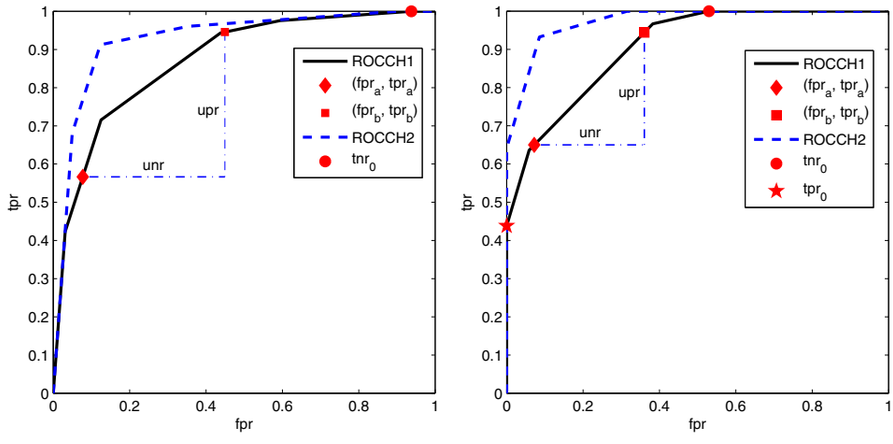
5.4.1 Types of ROC Isometrics

The main tools of our approach to construct reliable classifiers are the so-called *ROC isometrics*, which are defined as curves in ROC space that connect points with the same value for a (skew-sensitive) performance metric.



(a) (b)

Figure 5.3: Dominating abstention ROC curve (dashed curve) in a specific region of ROC space: (a) application of Theorem 5.3, and (b) application of Theorem 5.4.



(a) (b)

Figure 5.4: Dominating abstention ROC curve (dashed curve) on the complete covered part: (a) application of Theorem 5.6, and (b) application of Corollary 5.2.

For now, we consider the following three metrics: (1) precision, (2) F-measure, and (3) m-estimate. Accuracy will be considered later. Each metric has a positive variant and a negative variant. The positive variant measures performance on the positive classifications and the negative variant measures performance on the negative classifications. Hence, in total we consider six different metrics. Table 5.1 summarises them and Table 5.2 shows the definition of the corresponding isometrics. Each of these isometrics is obtained by rewriting the equation of the corresponding metric to that of a curve in ROC space. For generality and ease of representation, the isometric of the positive (negative) variant of a performance metric is called a positive (negative) isometric.

We note that all isometrics that we consider are linear curves (lines). Varying the skew ratio or the performance value results in a rotation of the isometric around a single point in which the performance metric is undefined. We will now investigate the isometrics of the aforementioned six performance metrics in more detail.

Precision

Positive precision is defined as the proportion of true positives to the total number of positive classifications:

$$prec_p^c = \frac{TP}{TP + FP} = \frac{tpr}{tpr + c\, fpr} \quad . \tag{5.6}$$

It is easy to show that the corresponding isometrics are lines that rotate around point $(0, 0)$. Also, according to our intuition, the higher the skew ratio and/or the positive precision value, the more the isometric approaches the optimal point $(0, 1)$.

Similar to the definition of positive precision, we define negative precision as the proportion of true negatives to the total number of negative classifications:

$$prec_n^c = \frac{TN}{TN + FN} = \frac{tnr}{tnr + \frac{1}{c}\, fnr} \quad . \tag{5.7}$$

Its isometrics rotate around point $(1, 1)$. It is easy to verify that lower skew ratios and/or higher negative precision values result in isometrics approaching the optimal point $(0, 1)$.

Figure 5.5(a) shows both type of isometrics for $c = 1$. In this and subsequent figures, we vary the value of the performance metric from 0.1 to 0.9 in steps of 0.1. We note that the described movement of the precision isometrics with regard to the skew ratio and performance value also holds for the other isometrics that we will describe below.

F-measure

Positive precision is maximised when all positive classifications are correct. Hence, maximal positive precision is trivially obtained by correctly classifying a single evidently positive instance while making no other positive predictions (so, we have one true positive and zero false positives). Clearly, it would be better to combine

Table 5.1: The six skew-sensitive performance metrics defined in terms of *fpr*, *tpr*, $c = N/P$, $\alpha \in \mathbb{R}^+$, and $\hat{m} = m/(P+N)$. For each of the conventional metrics we introduce a positive variant and a negative variant.

Metric	Indicator	Formula
Positive precision	$prec_p^c$	$\frac{tpr}{tpr+c\,fpr}$
Negative precision	$prec_n^c$	$\frac{tnr}{tnr+\frac{1}{c}fnr}$
Positive F-measure	$F_p^{c,\alpha}$	$\frac{\left(1+\alpha^2\right)tpr}{\alpha^2+tpr+c\,fpr}$
Negative F-measure	$F_n^{c,\alpha}$	$\frac{\left(1+\alpha^2\right)tnr}{\alpha^2+tnr+\frac{1}{c}fnr}$
Positive m-estimate	$mest_p^{c,\hat{m}}$	$\frac{tpr+\hat{m}}{tpr+c\,fpr+\hat{m}(1+c)}$
Negative m-estimate	$mest_n^{c,\hat{m}}$	$\frac{tnr+\hat{m}}{tnr+\frac{1}{c}fnr+\hat{m}\frac{1+c}{c}}$

Table 5.2: The corresponding ROC isometrics defined in terms of *fpr*, *tpr*, $c = N/P$, $\alpha \in \mathbb{R}^+$, and $\hat{m} = m/(P+N)$. Each of the isometrics is defined by a skew ratio, a value for the underlying performance metric, and additional parameters (if any) inherent to the performance metric.

Metric	Isometric
Positive precision	$tpr = \frac{prec_p^c}{1-prec_p^c}\ c\,fpr$
Negative precision	$tpr = \frac{1-prec_n^c}{prec_n^c}\ c\,fpr + 1 - \frac{1-prec_n^c}{prec_n^c}\ c$
Positive F-measure	$tpr = \frac{F_p^{c,\alpha}}{1+\alpha^2-F_p^{c,\alpha}} c\,fpr + \frac{\alpha^2\ F_p^{c,\alpha}}{1+\alpha^2-F_p^{c,\alpha}}$
Negative F-measure	$tpr = \frac{1+\alpha^2-F_n^{c,\alpha}}{F_n^{c,\alpha}} c\,fpr + 1 + \frac{(1+\alpha^2)(F_n^{c,\alpha}-1)}{F_n^{c,\alpha}} c$
Positive m-estimate	$tpr = \frac{mest_p^{c,\hat{m}}}{1-mest_p^{c,\hat{m}}} c\,fpr + \frac{\hat{m}\left(mest_p^{c,\hat{m}}(1+c)-1\right)}{1-mest_p^{c,\hat{m}}}$
Negative m-estimate	$tpr = \frac{1-mest_n^{c,\hat{m}}}{mest_n^{c,\hat{m}}} c\,fpr + 1 - \frac{1-mest_n^{c,\hat{m}}}{mest_n^{c,\hat{m}}} c + \frac{\hat{m}\left(mest_n^{c,\hat{m}}(1+c)-c\right)}{mest_n^{c,\hat{m}}}$

the metric with the true positive rate in order to know whether sufficient positive instances are used such that the metric can be considered as useful/accurate. For this reason, the positive F-measure was introduced by Van Rijsbergen (1979) as

$$F_p^{c,\alpha} = \frac{(1+\alpha^2)\ prec_p^c\ tpr}{\alpha^2\ prec_p^c + tpr} = \frac{\left(1+\alpha^2\right) tpr}{\alpha^2 + tpr + c\ fpr} \ , \tag{5.8}$$

where parameter $\alpha \in \mathbb{R}^+$ indicates the importance given to $prec_p^c$ relative to tpr as follows. We have that tpr is less (more) important than $prec_p^c$ when $\alpha < 1$ ($\alpha > 1$). They are considered equally important when $\alpha = 1$. The positive F-measure has the interesting property that it is high only when both $prec_p^c$ and tpr are high.[2] Its isometrics are lines that rotate around $(-\alpha^2/c, 0)$ and therefore they can be seen as a shifted version of the $prec_p^c$-isometrics. By inspecting this relationship more closely, we see that the larger c and/or the smaller α, the smaller the difference becomes between $F_p^{c,\alpha}$-isometrics and $prec_p^c$-isometrics.

Negative precision has a similar disadvantage as positive precision, i.e., it can be trivially maximised. Analogously, the negative F-measure is used for the trade-off between $prec_n^c$ and tnr:

$$F_n^{c,\alpha} = \frac{(1+\alpha^2)\ prec_n^c\ tnr}{\alpha^2\ prec_n^c + tnr} = \frac{\left(1+\alpha^2\right) tnr}{\alpha^2 + tnr + \frac{1}{c}\ fnr} \ , \tag{5.9}$$

where α now indicates the importance given to $prec_n^c$ relative to tnr. Corresponding isometrics are shifted versions of the $prec_n^c$-isometrics and rotate around $(1, 1 + \alpha^2 c)$. The smaller c and/or the smaller α, the less difference we observe with $prec_n^c$-isometrics. Figure 5.5(b) shows the isometrics for $c = 1$ and $\alpha = 1$.

m-estimate

The m-estimate computes a precision assuming that m "imagined" instances are classified a priori. It follows that the metric is less sensitive to noise and more effective in avoiding overfitting than precision is (Lavrac and Dzeroski, 1994, Ch. 8-10). This is in particular true when the class distribution is highly skewed and the classification performance for the minority class is measured. More specifically, by including a prior, we correct for an overly optimistic performance indicator when too few data are available (Fürnkranz and Flach, 2005).

The positive m-estimate assumes that the m instances are a priori classified as positive. They are distributed according to the class distribution, so some of them are positive instances while others are negative ones (and thus incorrectly classified). This gives the following definition:

$$mest_p^{c,m} = \frac{TP + m\frac{P}{P+N}}{TP + FP + m} = \frac{tpr + \frac{m}{P+N}}{tpr + c\,fpr + \frac{m}{P}} \ . \tag{5.10}$$

[2] Note that $prec_p^c$ and tpr are typically antagonistic: if $prec_p^c$ goes up then tpr usually goes down, and vice versa.

To eliminate the absolute numbers P and N, we define $\hat{m} = m/(P+N)$ and obtain the formula as given in Table 5.1. Corresponding isometrics rotate around $(-\hat{m}, -\hat{m})$ and they converge to the positive precision isometrics when $\hat{m} \to 0$. We can also show that, when $\hat{m} \to \infty$, the m-estimate converges to $1/(1+c)$, which is equal to the probability that a randomly drawn instance is in fact a positive instance. The corresponding isometric is then the ascending diagonal.

The case of the negative m-estimate is in full analogy with the discussion for the positive variant. So, we have the following definition:

$$mest_n^{c,m} = \frac{TN + m\frac{N}{P+N}}{TN + FN + m} = \frac{tnr + \frac{m}{P+N}}{tnr + \frac{1}{c}\,fnr + \frac{m}{N}} \ , \tag{5.11}$$

and again we can rewrite to include $\hat{m}$. A second calculation shows that the rotation point of the isometrics is $(1+\hat{m}, 1+\hat{m})$. Finally, as we have done in the case of the positive variant, we can also derive conditions for the convergence of the negative variant isometric when $\hat{m}$ goes to zero or infinity. We refrain from presenting more details. Figure 5.5(c) shows the m-estimate isometrics for $c = 1$ and $\hat{m} = 0.1$.

5.4.2 Overview of the ROC Isometrics Approach

Our approach departs from a scoring classifier and its ROC curve (more precisely, the ROCCH, but we will again simply write curve in the following). The inputs from the domain expert are the following: (1) the skew ratio, (2) the desired performance on the positive classifications, and (3) the desired performance on the negative classifications. Hence, given these values, a positive isometric and a negative isometric can be constructed. The intersection point of the positive isometric and the ROC curve represents by definition a classifier with the desired performance on the positive class. We recall that we denote this point by (fpr_a, tpr_a). Analogously, the intersection point (fpr_b, tpr_b) of the negative isometric and the ROC curve represents a classifier with the desired performance on the negative class. Finally, the intersection point of the isometrics themselves represents the desired reliable classifier.

Depending on the location of the reliable classifier in ROC space, we distinguish three cases as shown in Fig. 5.6. Each case needs a separate treatment as follows.

- **Case 1:** the isometrics intersect on the ROC curve.
  It follows from Theorem 2.1 that the desired classifier can be constructed by a single threshold that is applied on the score of instances. Hence, the reliable classifier is obtained by transforming the scoring classifier into a discrete classifier for which the appropriate threshold is found by a simple table look-up.

- **Case 2:** the isometrics intersect below the ROC curve.
  Theorem 2.1 also applies in this case. However, classifiers corresponding to points on the ROC curve between (fpr_b, tpr_b) and (fpr_a, tpr_a) have a higher performance on both classes. Clearly, one of these classifiers should be preferred since we are not interested in downgrading classification performance.

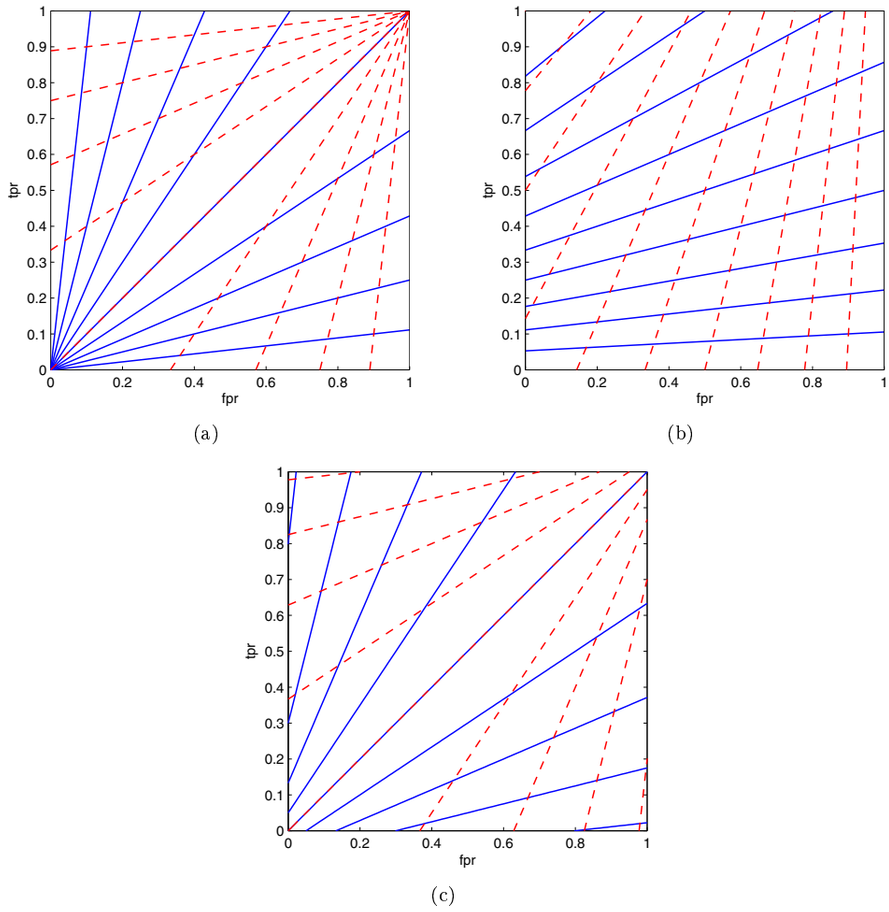
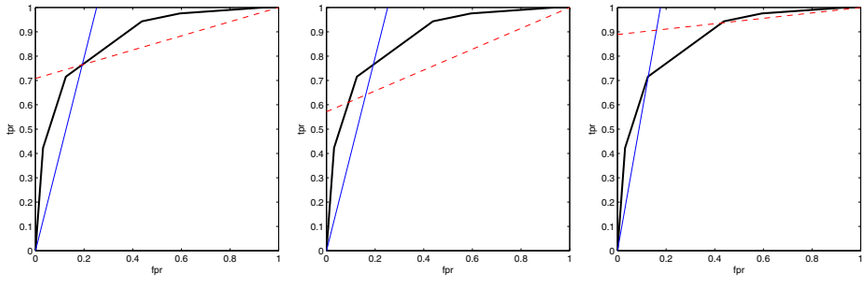


Figure 5.5: Visualisation of the ROC isometrics. The dashed (solid) curves represent the isometrics for the positive (negative) variant of the following classification performance metrics: (a) precision, (b), F-measure, and (c) m-estimate.



(a) (b) (c)

Figure 5.6: Location of the intersection point between a positive isometric and a negative isometric: (a) Case 1, (b) Case 2, and (c) Case 3.

- **Case 3:** the isometrics intersect above the ROC curve.
  The reliable classifier cannot be constructed using a single threshold. Our proposed solution is to define an abstaining classifier to filter out instances for which there is uncertainty in the true class label. The thresholds $a > b$ of the abstaining classifier are identified to correspond with points (fpr_a, tpr_a) and (fpr_b, tpr_b), respectively. When there is more than one intersection point for the positive (negative) isometric and the ROC curve, we choose the one with highest tpr (lowest fpr) such that $fpr_a < fpr_b$. This implies that the number of unclassified instances is always minimised. We can use Theorem 5.1 to construct the abstention curve. Consequently, the intersection points are transformed to the same point $(fpr'_a, tpr'_a) = (fpr'_b, tpr'_b)$. In the next subsection, we show that the type of isometrics (precision, F-measure, and m-estimate) defines the effect on the performance of the classifier corresponding to (fpr'_a, tpr'_a). We conjecture that this is a reliable classifier.

The three cases that we distinguished cover all possible scenarios that can occur. Cases 1 and 2 show that the reliable classifier can easily be constructed since the desired performances as defined by the domain expert can be obtained by a non-abstaining classifier (finding the appropriate threshold is a simple table look-up procedure). An interpretation of case 3 is not straightforward. Therefore, in the next subsection, we present an analysis to verify whether the proposed abstaining classifier is indeed a reliable classifier.

5.4.3 Formal Analysis of the Approach

We provide a separate analysis for each performance metric since different performance metrics imply different isometrics. As before, the proofs of the theorems are excluded from the main text and gathered in the appendix.

Precision

Theorem 5.7 shows the existence of a reliable classifier when the desired performance is represented by precision on each class. Theorem 5.8 shows that the ROC isometrics approach can also construct a reliable classifier with a preset accuracy. This is achieved by constructing two precision isometrics with the desired accuracy as the performance value.

Theorem 5.7. *If the points* (fpr_a, tpr_a) *and* (fpr_b, tpr_b) *are defined by a* $prec_p^c$*-isometric and* $prec_n^c$*-isometric respectively, then the point* (fpr'_a, tpr'_a) *has the precisions of both isometrics.*

Theorem 5.8. *If point* (fpr'_a, tpr'_a) *has* $prec_p^c = prec_n^c$*, then the accuracy in this point equals the precisions.*

Note that Theorem 5.8 implies that the ROC isometrics approach can guarantee a preset accuracy on each class, and the cost distribution as well as the class distribution can be incorporated via the skew ratio. Therefore, the approach overcomes the two problems of the conventional performance evaluation by means of accuracy, as explained in Subsection 5.2.2. Also, from the proof of the theorem it follows that, if the precisions of the isometrics are not equal, then the accuracy is bounded by the smallest precision and the largest precision.

F-measure

Theorem 5.9 shows the existence of a reliable classifier when the preset performance is represented by the F-measure on each class. In fact, the classifier corresponding to (fpr'_a, tpr'_a) has higher F-measures than the ones defined by the isometrics. This implies that the abstaining classifier exceeds the preset requirements, and therefore the classifier is clearly reliable.

Theorem 5.9. *If points* (fpr_a, tpr_a) *and* (fpr_b, tpr_b) *are defined by an* $F_p^{c,\alpha}$*-isometric and* $F_n^{c,\alpha}$*-isometric respectively, then the point* (fpr'_a, tpr'_a) *has at least the* F*-measures of both isometrics.*

Figure 5.7 gives an example where the obtained positive F-measure is approximately 5% higher than the desired performance on the positive class (i.e., than the performance of the positive isometric). The obtained negative F-measure is approximately 10% higher than the desired performance on the negative class.

It is an open question how to relate the preset performance values and the ones used in the construction of the isometrics such that the resulting classifier has exactly the preset performances (and not at least).

m-estimate

An analysis of the ROC isometrics approach using m-estimate isometrics becomes more subtle. After the transformation to an abstaining classifier, we can consider two cases: (1) the number of a priori classified instances m is kept fixed, and (2) the

parameter $\hat{m} = m/(P+N)$ is kept fixed. For brevity and readability, we will now analyse both cases in an intuitive manner. More details and technical considerations are found in the proofs of the theorems that we present below.

First, we consider the case where m is kept fixed. In this case, *upr* and *unr* can change the distribution of a priori instances over the classes. Intuitively, if $upr < unr$ then the distribution of these imagined instances in the positive m-estimate moves to the true positives resulting in a higher performance. For the negative m-estimate, the distribution moves to the false negatives resulting in a lower performance. The case of $upr > unr$ is the other way around. Therefore, an increase in performance on both classes is only possible when $upr = unr$.

Second, we consider the case where $\hat{m}$ is kept fixed. This implies that the distribution of a priori instances over the classes is left unchanged after transformation, although absolute numbers did change. A similar reasoning as done in the first case, and taking into account some extra difficulties, results in an improvement of the positive m-estimate if $upr \leq unr$ and $tpr_a \geq fpr_a$. The latter condition holds for all points on the convex hull of the ROC curve. Analogously, improvement in the negative m-estimate occurs if $upr \geq unr$ and $tpr_b \geq fpr_b$. Thus, summarising our results, we arrive at the following two theorems.

Theorem 5.10. *If point (fpr_a, tpr_a) is defined by a $mest_p^{c,\hat{m}}$-isometric with $m > 0$ and if $upr \leq unr$, then the point (fpr'_a, tpr'_a) has at least the positive m-estimate of that isometric.*

Theorem 5.11. *If point (fpr_b, tpr_b) is defined by a $mest_n^{c,\hat{m}}$-isometric with $m > 0$ and if $upr \geq unr$, then the point (fpr'_a, tpr'_a) has at least the negative m-estimate of that isometric.*

Corollary 5.3. *If points (fpr_a, tpr_a) and (fpr_b, tpr_b) are defined by a $mest_p^{c,\hat{m}}$-isometric and $mest_n^{c,\hat{m}}$-isometric respectively with $m > 0$ and $upr = unr$, then the point (fpr'_a, tpr'_a) has at least the m-estimates of both isometrics.*

From the corollary follows that a reliable classifier is obtained when the rate of abstention on the positive class and on the negative class are equal. This condition is difficult to satisfy, and to relax it, we propose to use the m-estimate for the minority class and the usual precision for the majority class. Indeed, from Theorems 5.10 and 5.11 it follows that we need an abstention characterised by $upr \leq unr$ ($upr \geq unr$) when the minority class is the positive (negative) class. This condition is likely to be satisfied in domains with a skewed class distribution since the m-estimate isometric of the minority class has to cover a large part in ROC space in order to use sufficient data for an accurate performance indication. Figure 5.8 shows an example with fixed m and the negative class as minority class. The $mest_n^{c,\hat{m}}$-isometric has a low slope such that sufficient negative instances are covered for the metric to be accurate. Consequently, the condition $upr \geq unr$ is easily satisfied and a reliable classifier is obtained.

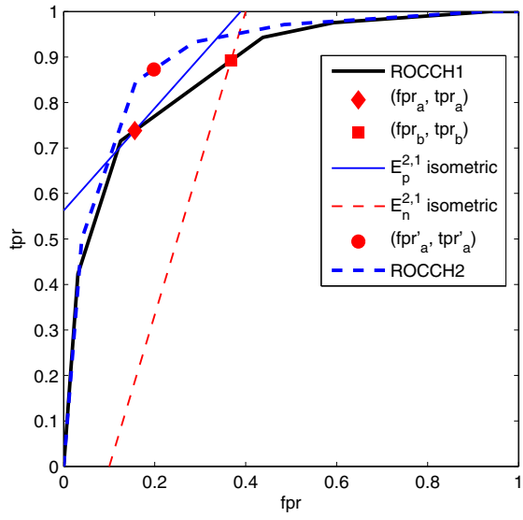


Figure 5.7: The ROC isometrics approach for the F-measure: $F_p^{2,1} = 0.72$ in (fpr_a, tpr_a) and $F_n^{2,1} = 0.75$ in (fpr_b, tpr_b). The reliable classifier has $F_p^{1.84,1} = 0.77$ and $F_n^{1.84,1} = 0.86$. The abstention rates are $upr = 0.15$ and $unr = 0.21$.

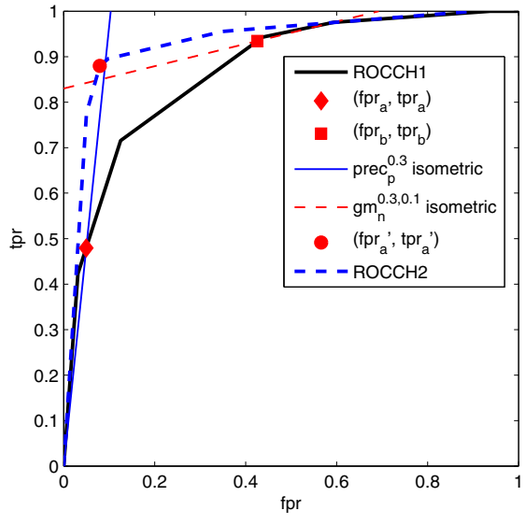


Figure 5.8: The ROC isometrics approach for the m-estimate: $prec_p^{0.3} = 0.97$ in (fpr_a, tpr_a) and $mest_n^{0.3,0.1} = 0.55$ in (fpr_b, tpr_b). The reliable classifier has $prec_p^{0.3} = 0.97$ and $mest_n^{0.34,0.18} = 0.56$. The abstention rates are $upr = 0.45$ and $unr = 0.38$.

5.4.4 Conclusions from the Theoretical Results

The formal analysis of the ROC isometrics approach was divided into three separate cases. Cases 1 and 2 are almost trivial since the preset performance is at most as high as the performance of a conventional discrete classifier. Case 3 is the most interesting one since it uses an abstaining classifier in order to improve on the maximum classification performance obtainable so far.

We showed formally that also the third case leads to a reliable classifier for the performance metrics that we have considered. The number of unclassified instances is the lowest possible. The only requirement to construct the classifier is a positive isometric and a negative isometric that intersect the (convex) ROC curve. The ROC isometrics approach does not assume that the classifiers produce probabilities, i.e., any measure of confidence in the positive class can be used for the scores. This is in contrast to ensemble methods for which the scores often should be true conditional class probabilities (Cohen and Goldszmidt, 2004). Other disadvantages of ensemble methods, including the popular boosting method, is that there is no guarantee to which level the performance can be increased and the resulting models are not comprehensible. Thus, from our formal analysis, we may conclude that the ROC isometrics approach is generally applicable, and it is a simple, effective, and efficient answer to construct reliable classifiers.

Finally, we comment on the implicit assumption of our formal analysis, namely that the empirical ROC curve is an accurate estimate of the true curve. This assumption can be verified by deriving confidence bands along the curve (Macskassy and Provost, 2004). In essence, the lower band and the upper band can also be used as a guideline to adjust the thresholds when the assumption does not seem to hold. However, from our experimental analysis given in the next section, we found that even for small data sets the empirical curve is sufficiently accurate. Recently introduced generalisation bounds for the area under the ROC curve give further evidence for this observation (Agarwal *et al.*, 2005).

5.5 Experimental Analysis of the Approach

In this section, we provide an empirical evaluation of the ROC isometrics approach. The approach is tested by applying two popular classifiers on ten benchmark data sets (5.5.1). We outline the experimental setup (5.5.2), discuss the results (5.5.3), and we formulate empirical conclusions (5.5.4).

5.5.1 Data Sets and Classifiers

We tested the ROC isometrics approach on six binary classification data sets from the UCI benchmark repository (Asuncion and Newman, 2007) and four larger data sets from a recent machine-learning competition (Guyon, 2007). As a pre-processing step, all instances with missing feature values were removed as well as duplicate instances. Features were also standardised to have zero mean and unit variance. We

Table 5.3: The ten data sets, where the column headings refer to: (1) reference number, (2) data set name, (3) number of instances, (4) minority class, (5) percentage of examples in the minority class, (6) accuracy of k-NN, and (7) accuracy of NB.

#	name	size	min. class	% min. class	acc. k-NN	acc. NB
1	heart statlog	270	+1	44.44	84.81%	85.93%
2	house votes	342	+1	34.21	94.13%	94.49%
3	ionosphere	350	−1	35.71	90.57%	90.29%
4	monks3	432	−1	48.15	78.70%	83.10%
5	sonar	208	+1	46.63	89.90%	89.82%
6	spect	219	−1	12.79	88.12%	89.50%
7	ada	4562	+1	24.81	83.55%	83.43%
8	gina	3468	+1	49.16	93.31%	93.11%
9	hiva	4229	+1	3.52	97.22%	97.89%
10	sylva	14394	+1	6.15	97.57%	97.74%

have summarised the main characteristics of the resulting data sets in Table 5.3. The reported accuracies are obtained by a stratified ten-fold cross-validation procedure.

Our first classifier is the k-nearest neighbour classifier (k-NN) which classifies an instance by means of a majority vote among the labels of its k nearest neighbours (Cover and Hart, 1967). To convert the classifier into a scoring classifier, we defined the score of an instance $\boldsymbol{x}$ as $\sum_{l=1}^{k} d_l^n / \sum_{l=1}^{k} d_l^p$ where d_l^n and d_l^p are the distances from $\boldsymbol{x}$ to its l-th neighbour with negative and positive class label, respectively. Clearly, the score is valid since it increases when distances to nearest neighbours of the positive class decrease and/or distances to nearest neighbours of the negative class increase. Nearest neighbours are found by Euclidean distances. The value of k is restricted to $\{1, 2, \ldots, 10\}$ and is chosen using an independent validation set of size 20% of the training set. The combination of k-NN and the ROC isometrics approach is called ROC-kNN.

Our second classifier is the Naive Bayes classifier (NB), which is a probabilistic classifier that applies Bayes theorem with independence assumptions (Domingos and Pazzani, 1997). NB outputs an estimate of the conditional class probability $\mathbb{P}(p \,|\, \boldsymbol{x})$, and clearly, this value can be used as a score without modification. Continuous features are handled by a simple method of binning (Lu, Yang, and Webb, 2006). We have chosen for ten bins. Henceforth, we use ROC-NB to denote the combination of NB and the ROC isometrics approach.

5.5.2 Experimental Setup

The classifiers ROC-kNN and ROC-NB are applied on the ten benchmark data sets using a stratified ten-fold cross-validation procedure. This procedure was repeated for hundred times with different random permutations of the data in order to obtain robust results for the smaller data sets. Each training fold is used to construct an

ROC curve in order to find the threshold(s). The isometrics are based on the precision metric and we set the positive precision equal to the negative precision. Hence, we have that the precision value is equal to the accuracy value (see Theorem 5.8). The other performance metrics implied results that led to similar conclusions than those derived from accuracy, so we refrain from presenting details.

Quality assessment of ROC-kNN and ROC-NB is done by measuring and reporting two key statistics of the classifiers. First, we measure the average percentage of correct classifications over all test folds. This is the *empirical performance* and its value should be at least the preset performance. Second, we measure the average percentage of instances for which a label is predicted. This statistic is called the *efficiency* and its value represents how useful the classifier is when applied in practice. A large efficiency is desired since it is inversely related to abstention rate.

In the experiments we consider five preset performances that we believe to be of interest in many classification tasks: 95%, 96%, 97%, 98%, and 99%. We do not consider 100% preset performance since we found that, although some ROC curves have a large area under the curve, there is no vertical first line segment and/or no horizontal last line segment. This implies that at least one negative instance (positive instance) is assigned the highest (lowest) score. We therefore advice to pre-process the data to remove noise and outliers whenever the approach is used in a real-life application. For the purpose of benchmarking this is however not desired.

5.5.3 Experimental Results

We now report and compare the experimental results of the classifiers ROC-kNN and ROC-NB. The results are given in Tables 5.4 to 5.6. We start by giving two remarks on these tables. First, results on the `hiva` and `sylva` data sets are omitted for preset performances below 98% since this is the performance of the original classifier (see again Table 5.3). Second, omitted values marked by an asterisk indicate that the preset performance did not result in two intersection points with the ROC curve.[3] Except for the `spect` data set, asterisks only occur for ROC-NB which implies that the NB score is not as discriminative as the k-NN score.

Table 5.4 shows the empirical performances. We see that the values are equal to the preset performances up to statistical fluctuations, even for the small data sets. These results verify that the classification performance can be preset by a domain expert. Therefore, the classifiers are reliable classifiers. We note that the statistical fluctuations arise from averaging values over many test folds. So admittedly, when the number of processed instances becomes small, there is a larger gap between empirical and preset performance. This is however also the case in related work; see the next section for an overview and discussion.

Table 5.5 shows the differences between the positive class performances and the negative class performances. Since some data sets have a highly unbalanced class distribution, it is desired that these performances are approximately equal. As expected, the differences for ROC-kNN and ROC-NB show that the preset performance

[3] In essence, it is sufficient that at least one isometric intersects the curve. We do not consider this setup in our experiments since it leads to a classifier that always predicts the same class label.

approximately holds for both classes, with the exception of `ionosphere`, `monks3`, and `ada`. For these data sets, the classifiers seem to suffer from statistical fluctuations. This is especially the case for `ada` where the sign of the difference shows that a bad performance on the positive class (the minority class) is masked by a good performance on the negative class.

We explain the larger differences by a randomisation of thresholds. More specifically, when an isometric intersects the convex ROC curve in an endpoint of two adjacent line segments, then the corresponding threshold can be found directly from the scores of the instances that are used to construct the curve. Otherwise, a randomisation of two thresholds on scores is needed, as shown in the proof of Theorem 2.1. In the case of intersection with a large line segment, this can result in a larger deviation from preset performance when not sufficient instances are being processed. Our explanation is verified in Table 5.6 since, in general, larger deviations occur when the difference in empirical performances per class is large. We also found that the largest deviations occurred when the number of distinct scores produced by the classifiers is small. This then again results in less and larger line segments.

Finally, we show the efficiency of ROC-kNN and ROC-NB in Table 5.7. We see that, in general, the efficiency declines exponentially when the preset performance is increased. In addition, comparing the two classifiers directly, ROC-kNN can be claimed as the most efficient one. This is the case since the k-NN scores lead to ROC curves that tend more toward the optimal point $(0, 1)$ in ROC space than the NB scores. Since different data sets and classifiers lead to different convex hulls, it is advantageous to plot the performance increase relative to the non-abstaining classifier as a function of the abstention rate. Such graphs provide a support tool for the domain expert who might like to decide on a trade-off between preset performance and abstention rate. Figure 5.9 shows such graphs for the `heart statlog` and `gina` data sets.

5.5.4 Conclusions from the Experimental Results

The experimental results are in line with the conclusions that we derived from the theoretical analysis. More specifically, the results show that reliable classifiers can be constructed with a preset performance that is at least the performance of the original classifier (up to statistical fluctuations due to averaging performance values obtained from finite data sets). The efficiency of such a reliable classifier depends on the shape of the original ROC curve, or in other words, on the discriminative power of the scores produced by the scoring classifier. When a badly structured ROC curve is given as input to the approach, some preset performances may not be achievable since the isometrics do not intersect the curve. In our experiments, this occurred only a few times; in general, when the preset performance is almost 100%. For use in real applications, the problem can be alleviated or avoided by pre-processing the data to detect outliers and other anomalies.

Table 5.4: Empirical performances obtained by ROC-kNN (left part) and ROC-NB (right part): empirical performances are identical to preset performances up to statistical fluctuations, even for the small data sets.

Data set	ROC-kNN					ROC-NB				
	95%	96%	97%	98%	99%	95%	96%	97%	98%	99%
1	94.5	95.6	97.0	98.0	98.8	93.5	95.0	96.1	97.5	98.0
2	95.8	96.1	97.2	97.7	98.7	95.0	96.4	97.0	97.7	98.7
3	95.1	95.0	96.1	97.5	98.6	94.6	95.3	*	*	*
4	95.6	96.0	96.9	97.7	99.0	94.6	95.3	96.7	97.1	99.9
5	94.8	95.9	97.0	98.1	98.8	94.1	95.9	97.3	98.1	*
6	*	*	*	*	*	*	*	*	*	*
7	96.0	96.9	97.8	98.5	99.3	96.0	96.9	98.4	*	*
8	95.0	96.0	97.0	98.0	99.0	94.5	95.4	96.8	97.7	98.7
9	-	-	-	98.0	99.0	-	-	-	97.9	99.1
10	-	-	-	98.0	99.0	-	-	-	98.0	99.0

Table 5.5: Difference between the empirical performances on the positive class and the negative class by applying ROC-kNN (left part) and ROC-NB (right part): in general, empirical performances are balanced over the classes.

Data set	ROC-kNN					ROC-NB				
	95%	96%	97%	98%	99%	95%	96%	97%	98%	99%
1	0.2	1.8	-2.5	-2.5	-2.0	-1.2	-3.5	-3.7	-3.8	-6.7
2	-2.1	-1.7	-1.5	-0.1	-2.1	-2.2	-1.2	-1.0	-0.7	-2.8
3	-1.7	-1.9	-7.4	-10.0	-8.3	-5.7	-4.6	*	*	*
4	5.0	7.8	6.9	1.3	1.2	6.0	5.7	0.2	-0.5	0.0
5	-0.2	-0.1	-2.2	1.1	-1.3	-0.5	1.0	1.1	-0.9	*
6	*	*	*	*	*	*	*	*	*	*
7	-5.0	-6.5	-4.5	-5.0	-6.9	-10.1	-9.1	-7.4	*	*
8	0.0	0.0	-0.2	-0.5	-0.6	-1.6	-1.1	-0.4	-0.3	-2.2
9	-	-	-	-0.6	-1.3	-	-	-	-3.4	-4.7
10	-	-	-	-0.2	0.1	-	-	-	-1.2	-1.1

Table 5.6: Standard deviations of the empirical performance by applying ROC-kNN (left part) and ROC-NB (right part): randomisation of thresholds has often minor influence, even for the small data sets.

Data set	ROC-kNN					ROC-NB				
	95%	96%	97%	98%	99%	95%	96%	97%	98%	99%
1	0.77	0.44	1.08	0.95	1.03	1.35	2.04	1.59	1.99	1.35
2	0.44	0.36	0.26	0.51	0.23	0.47	0.53	0.57	0.34	0.32
3	0.33	0.33	0.42	1.35	0.40	0.74	1.12	*	*	*
4	0.76	0.62	0.65	0.44	0.44	0.60	0.56	0.54	0.47	0.44
5	0.34	0.65	0.45	0.60	0.43	0.60	0.72	0.72	0.66	*
6	*	*	*	*	*	*	*	*	*	*
7	0.88	0.92	1.35	0.90	1.17	1.32	0.98	1.38	*	*
8	0.07	0.06	0.03	0.04	0.06	0.14	0.20	0.23	0.21	0.11
9	-	-	-	0.09	0.02	-	-	-	0.11	0.11
10	-	-	-	0.02	0.01	-	-	-	0.07	0.08

Table 5.7: Efficiency of ROC-kNN (left part) and ROC-NB (right part): ROC-kNN is the most efficient reliable classifier on the majority of the data sets.

Data set	ROC-kNN					ROC-NB				
	95%	96%	97%	98%	99%	95%	96%	97%	98%	99%
1	55.6	54.0	47.0	40.0	35.0	43.3	37.0	30.0	27.4	26.3
2	99.4	99.4	97.6	95.9	87.9	98.5	95.9	95.6	93.3	83.0
3	84.0	78.0	52.0	35.7	30.0	80.0	74.0	*	*	*
4	57.2	53.3	52.8	51.2	44.2	60.5	56.7	55.4	51.9	463
5	90.0	84.5	79.5	77.5	72.5	82.5	80.5	73.5	70.0	*
6	*	*	*	*	*	*	*	*	*	*
7	42.6	37.8	31.9	25.2	19.6	38.1	35.3	18.5	*	*
8	96.5	93.8	89.5	83.8	67.2	93.5	88.4	83.5	79.0	61.8
9	-	-	-	80.0	73.6	-	-	-	75.8	73.5
10	-	-	-	99.5	96.6	-	-	-	97.9	95.0

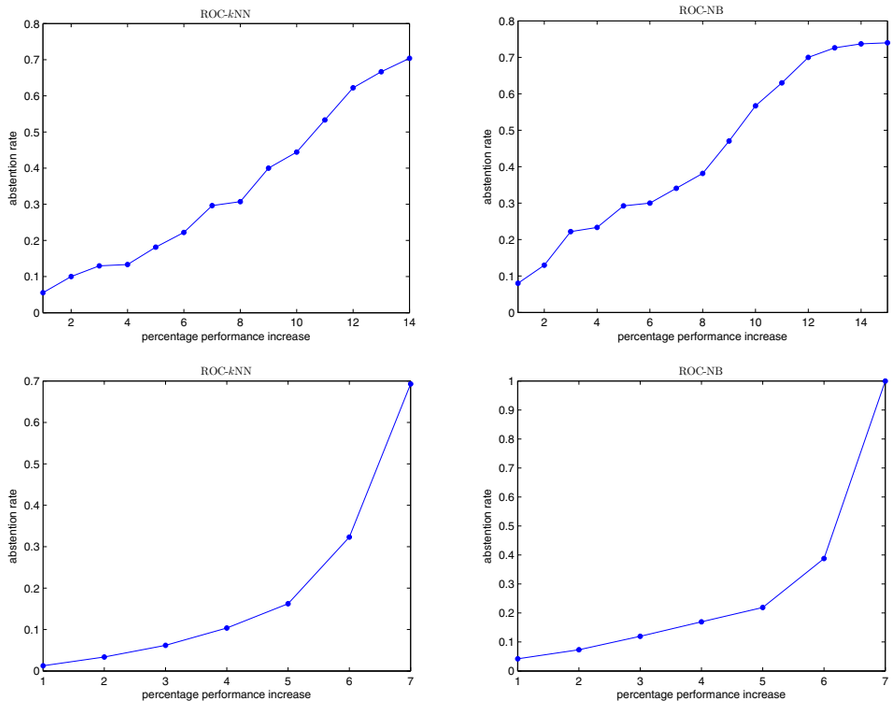


Figure 5.9: Abstention rate as a function of the performance increase relative to the non-abstaining classifier for ROC-kNN (left part) and ROC-NB (right part): the top row shows the results for `heart statlog` and the bottom row for `gina`. Performance is increased in steps of 1% on the horizontal axis and the curves end at 100% preset performance.

5.6 Related Work

In this section, we give an overview of work about abstaining classifiers. We discuss analogies, advantages, and disadvantages with respect to the ROC isometrics approach.

5.6.1 Decision Theory

The first well-founded work on abstaining classifiers uses the framework of decision theory. The setting is a classification problem with m classes and the performance metric is expected cost, which is defined as

$$cost(\lambda_i) = \sum_{j=1}^{m} c(\lambda_i, \lambda_j)\mathbb{P}(\lambda_j \mid \boldsymbol{x}) = \sum_{j=1}^{m} c(\lambda_i, \lambda_j)\frac{\pi_j \mathbb{P}(\boldsymbol{x} \mid \lambda_j)}{\mathbb{P}(\boldsymbol{x})} \ ,$$

with $c(\lambda_i, \lambda_j)$ the cost for predicting class label λ_i when the true label is λ_j, π_j the class prior, $\mathbb{P}(\boldsymbol{x} \mid \lambda_j)$ the class likelihood, and $\mathbb{P}(\boldsymbol{x}) = \sum_{j=1}^{m} \pi_j \mathbb{P}(\boldsymbol{x} \mid \lambda_j)$ the marginal probability of observing instance $\boldsymbol{x}$.[4]

Assuming 0 cost for correct classifications, 1 cost for incorrect classifications, and t cost for abstentions ($0 < t < 1-1/m$), it can be shown that the abstaining classifier with lowest expected cost predicts the class with maximum posterior probability, $\mathbb{P}(\lambda_i \mid \boldsymbol{x})$, only if this value is at least $1-t$ (Chow, 1957; 1970). We can generalise this finding to arbitrary costs for incorrect predictions. Assuming a binary classification setting, $m = 2$, we can then show that the optimal abstaining classifier is defined by two thresholds. The values of these thresholds depend on the costs and the class priors; see Vanderlooy *et al.* (2009) for more details and a computation. The use of these thresholds is identical to what we have discussed so far.

Despite the benefit that the decision theory framework allows naturally for handling multi-class classification problems, its practical value is severely limited since it assumes that the distributions of the classes are perfectly known. In practice this is almost never the case. Careful experiments using synthetic data have shown that the resulting classifier has expected cost significantly higher than the lowest possible expected cost when noise is added to the class distributions (Fumera, Roli, and Giacinto, 2000; Marrocco, Molinara, and Tortorella, 2007).

5.6.2 ROC Analysis

Assume two points on an ROC curve corresponding to two thresholds $a > b$. Due to the strict ordering of these thresholds, we have that a true positive for the classifier corresponding to a is a true positive for the classifier corresponding to b. Analogously, a true negative for the classifier corresponding to b is a true negative for the classifier corresponding to a. So indeed, when the classifiers disagree, we actually abstain from

[4] Expected cost is equal to the expected error rate calculated according to a specific cost distribution. It can be seen as an alternative to our skew-sensitive accuracy metric.

classification. The expected cost, remembering that we can set benefits of correct predictions to zero, can therefore be defined in terms of measurements in ROC space:

$$cost(a,b) = \pi_p \, c(n,p) \, fnr_b + \pi_n \, c(p,n) \, fpr_a + \pi_p \, c(u,p) \, upr + \pi_n \, c(u,n) \, unr \ .$$

Rewriting results in a new equation in which the optimal values for a and b can be separately evaluated as follows (Tortorella, 2004; 2005). Each of the two evaluations defines the slope of a set of lines in ROC space. For each slope, we move the corresponding line from passing through the point $(0,1)$ to the lower right of ROC space until the line touches the ROC curve. The optimal threshold is then corresponding to the intersection point. It has been shown that this approach leads to the same abstaining classifier as the decision theory approach when there is complete knowledge of the class distributions (Santos-Pereira and Pires, 2005).

Compared with the ROC isometrics approach, there is a clear analogy since both rely on the slope of line segments of the ROC curve. Nonetheless, the above approach relies on a specific decomposition of the expected cost, which allowed to derive two slopes in a convenient way. It is not clear how this can be done for other metrics, if it is possible at all. In addition, the approach requires explicit knowledge of the error costs. It has been shown that these numbers have a large impact on the abstention rate, and correspondingly, it was concluded that the approach is difficult to apply in domains where the costs are not explicitly given (Pietraszek, 2007b).

To overcome these problems, two approaches have been proposed in which either the expected cost or the abstention rate is bounded by the domain expert (Pietraszek, 2007b). The first approach has the same motivation as we used for the ROC isometrics approach: guarantee a minimal preset performance while allowing as few abstentions as possible. A constraint on the performance is easily quantifiable and intuitive for the domain expert. The second approach is the inverse of the first one: find the classifier with minimum expected cost among all classifiers with abstention rate as most as high as a preset value. This approach is natural in resource-constrained situations where a human expert can only handle a limited number of events in some time period. Its usefulness has been validated for the application of intrusion detection (Pietraszek, 2007a). However, the approaches are restricted to expected cost and a closed-form solution does not always exist.

5.6.3 Prediction Sets

Instead of leaving an instance unclassified in case of uncertainty, it is also possible to output a prediction set, i.e., a set of class labels that contains the true class label with high confidence. Clearly, it is not wise to choose a priori a fixed size for the prediction sets. The size should depend on the particular instance to be classified and the error rate that one is willing to tolerate.

For some applications, it is beneficial to prefer prediction sets over unclassified instances. An example is the task of automated face identification (Wechsler, 2006). Law enforcement agencies are eager to apply such a classifier in public safety services that are based on surveillance cameras. With these cameras it is among others possible to scan public areas continuously and automatically in order to compare

the faces that a camera detects with the faces of wanted offenders stored in some database. The classifier's task is to predict whether the detected face corresponds to one of the wanted offenders. When the classifier is uncertain, it can reject the face and pass it to a human together with a set of candidate faces from the database. The human expert can easily verify whether one of the candidate faces indeed corresponds to the face detected by the camera. A second advantage of prediction sets is that they allow to differ between ignorance and uncertainty (Hüllermeier and Brinker, 2008). Ignorance corresponds to an empty prediction set: the classifier does not have sufficient information to predict any class label. Uncertainty means that several class labels could be the true one, resulting in a prediction set with two or more elements.

There has been some efforts in learning classifiers that produce prediction sets, and recently, these works are gaining importance. For example, the Bayesian framework was used to derive a simple classifier that is optimal in the sense that, for a given average number of classes in the prediction sets, no other classifier can have lower error rate (Ha, 1996a; 1996b; 1997). However, optimality is no longer guaranteed in practice when estimations of the true probabilities are used, and the error rate cannot be defined prior to classification. An extension of the Naive Bayes classifier to imprecise probabilities has also been proposed; see Corani and Zaffalon (2008) and references therein for more details. Hüllermeier (2004) focuses on the nearest neighbour classifier. The most widely used approach is that of transductive confidence machines and can be used for any classifier (Vovk, Gammerman, and Shafer, 2005). It does not depend on true probabilities and it is shown to be optimal with respect to error rate in a wide variety of learning settings. The approach tries out each possible class label for the test instance and approximates how likely it is that the resulting data have been generated by the underlying distribution. This results in p-values which are used to construct the prediction sets. An empirical comparison between the approach and the ROC isometrics approach in case of binary classification data sets, as well as a discussion about analogies and differences, is given by Vanderlooy and Sprinkhuizen-Kuyper (2007). It was concluded that the approaches are competing and promising generally applicable machine learning tools.

5.7 Chapter Conclusions

In the chapter, we focussed on the third research question (RQ 3): *Can we develop a feasible approach by which a classifier is constructed that guarantees a preset classification performance on each class?* We have answered this question by means of introducing the ROC isometrics approach. The approach is a successful combination of (1) classifier visualisation in ROC space, (2) isometrics, and (3) an abstaining classifier. We provided an analysis of the approach when precision, F-measure, and m-estimate are used to measure classification performance. In addition, our analysis showed that the approach can also be used with a cost-sensitive version of accuracy as well as with any combination of the aforementioned performance metrics. We empirically tested the approach by applying two popular classifiers on ten benchmark data sets. The experimental results confirmed the formal analysis.

Based on our theoretical analysis, the empirical evaluation, and our discussion about related work, we are able to formulate four conclusions. First, we may conclude that the ROC isometrics approach is generally applicable since any classifier can be used to construct an ROC curve. Some classifiers such as Naive Bayes and neural networks naturally provide scores. For other classifiers, such as nearest neighbour and support vector machine, a postprocessing technique may be needed or preferred (Cestnik, 1990; Atiya, 2005; Niculescu-Mizil and Caruana, 2005). Second, we may conclude that the ROC isometrics approach does not commit to specific cost distributions and class distributions since skew-sensitive isometrics are used. When there is a change in these distributions, a new reliable classifier is constructed efficiently from the original ROC curve by two straightforward steps: an update of the skew ratio in the isometrics, and a recalculation of the intersection points and thresholds. Third, we may conclude that the ROC isometrics approach is clearly efficient in terms of time complexity since it only involves the computation of intersection points and a table look-up. It is also efficient in terms of space complexity since it solely involves storing a convex hull and a short list of thresholds. Fourth, we may conclude that the approach is easy-to-visualise and our results on dominance relations provide guidelines to understand the effect of abstention.

In summary, we may state that the ROC isometrics approach provides a satisfactory answer to the third research question and opens new grounds for immediate applications in domains with high error costs such as the domain of law enforcement.

Chapter 6

Optimal Aggregation Strategy for Pairwise Classification

A common strategy to solve a multi-class problem is to decompose it into a series of binary classification problems and to learn a set of corresponding classifiers. The question is then how to aggregate the predictions of these base classifiers into a single final classification. In this chapter, we use the label ranking setting to derive under certain model assumptions a generalised voting strategy in which predictions are properly adapted according to the strength of the base classifiers. We call this strategy adaptive voting and show that it is optimal in the sense of yielding a maximum a posteriori (MAP) prediction of the class label of a test instance. Moreover, we show that the popular weighted voting yields a good approximation of the MAP prediction, and thus can be seen as a quasi-optimal aggregation strategy.[1]

6.1 Formal Strategy for Aggregating Predictions

So far we have considered binary classification problems. Although many problems are of this kind, sometimes the class label space consists of $m > 2$ classes. An example for the field of law enforcement is to classify offenders into several groups, where each group is assigned different penalties in order to reduce the risk of crimes in the future (Blokland *et al.*, 2005b). However, there exist various popular learning algorithms that cannot learn classifiers able to distinguish between more than two classes, e.g., logistic regression and support vector machine.

Learning by pairwise comparison is a well-known decomposition technique which allows to transform a multi-class classification problem into a number of binary problems (Fürnkranz, 2002). To this end, a separate classifier is trained for each pair of class labels. Typically, the technique produces more accurate models than the alternative one-against-rest decomposition, which learns one model for each class

[1] This chapter is based on an article by Hüllermeier and Vanderlooy (2009a) who extend earlier work in Hüllermeier and Vanderlooy (2008b).

using the examples of this class as positive examples and all others as negative examples. Also, despite the need to train a quadratic instead of a linear number of models, pairwise classification is computationally not more complex than one-against-rest (Fürnkranz, 2001; Fürnkranz, 2002; Hsu and Lin, 2002).

A critical step in pairwise learning is the aggregation of the predictions from the ensemble of binary models into a final classification. A large number of strategies have been proposed for this purpose. We refer to Friedman (1996), Moreira and Mayoraz (1998), Fürnkranz (2003), Wu, Lin, and Weng (2004) and Quost, Denoeux, and Masson (2007). Since the aggregation problem also occurs in all other decomposition methods and in ensemble methods, these research areas as well provide a large number of aggregation strategies (Xu, Krzyzak, and Suen, 1992). However, since the semantics of these problems are different, we note that the aggregation strategies from different fields are not always interchangeable. We solely focus on aggregating predictions from binary models learned by pairwise comparison.

After a close literature inspection, it becomes clear that the most commonly used aggregation strategy is the simple *weighted voting*. In this strategy, the prediction of each binary model is counted as a weighted "vote" for a class label and the class with the highest sum of votes is predicted. Even though weighted voting performs very well in practice, it is often criticized as being ad-hoc to some extent and for lacking a sound theoretical basis (Hüllermeier *et al.*, 2008; Hüllermeier and Fürnkranz, 2009).

Therefore, in this chapter, we are interested in answering our fourth research question (RQ 4): *Can we develop an aggregation strategy that is feasible in practice and shown to be optimal under reasonable conditions?* In this regard, we make three contributions. First, we propose a formal framework in which the aforementioned aggregation problem for pairwise learning can be studied in a convenient way. This framework is based on the setting of label ranking which has recently received attention in the machine learning literature (Har-Peled *et al.*, 2002; Crammer and Singer, 2003; Hüllermeier *et al.*, 2008). Second, within this framework, we develop a new aggregation strategy called *adaptive voting*. This strategy allows one to take the strength of individual classifiers into consideration and, under certain assumptions, is provably optimal in the sense that it yields a MAP prediction of the class label (i.e., it predicts the label with maximum posterior probability).[2] Third, we show that weighted voting can be seen as an approximation of adaptive voting and, thus, approximates the MAP prediction. This theoretical justification is complemented by arguments suggesting that weighted voting is quite robust toward incorrect predictions of the binary models.

All these results are confirmed by strong empirical evidence showing that adaptive voting is indeed able to outperform weighted voting in a consistent way, albeit by a very small margin. The experimental results also show that the superiority of adaptive voting only holds as long as its underlying model assumptions are (approximately) met by the ensemble of binary models. When assumptions are strongly violated, weighted voting is at least competitive and, in this sense, appears to be more robust than adaptive voting.

[2] We consider the *strength* of a classifier as its ability to separate classes with high confidence. We will give a precise definition in terms of a parametric probabilistic model in Subsection 6.3.1.

The remainder of the chapter is organised as follows. In Section 6.2 we review learning by pairwise comparison and the label ranking setting. We introduce the adaptive voting strategy and its underlying formal model in Section 6.3. In Section 6.4 we provide accompanying experiments with synthetic data. In Section 6.5 we focus on weighted voting and among others show that it can be seen as an approximation of adaptive voting. Experimental results on benchmark data sets are presented in Section 6.6. Finally, Section 6.7 concludes the chapter and provides our answer to the research question.

6.2 Pairwise Classification and Label Ranking

In this chapter, we focus on learning by pairwise comparison (6.2.1) and on the label ranking setting (6.2.2). After explaining these two concepts, below we also show how to predict label rankings (6.2.3) and we explain the benefits of using the label ranking setting for our formal analysis presented in this chapter (6.2.4).

6.2.1 Learning by Pairwise Comparison

Learning by pairwise comparison (also known as all-pairs, one-against-one, round robin learning, pairwise classification) is a popular decomposition technique that transforms an m-class classification problem, $m > 2$, into a number of binary problems (Fürnkranz, 2002). To this end, a separate model or base classifier $\mathcal{M}_{ij}$ is trained for each pair of labels $(\lambda_i, \lambda_j) \in \mathcal{L} \times \mathcal{L}$, $1 \leq i < j \leq m$; thus in total a number of $m(m-1)/2$ models is needed (see Fig. 6.1).

To train the base classifiers, an observation $(\boldsymbol{x}, \lambda_{\boldsymbol{x}})$, i.e., an instance $\boldsymbol{x}$ belonging to class $\lambda_{\boldsymbol{x}} = \lambda_k$, is submitted as a positive example $(\boldsymbol{x}, 1)$ to all $\mathcal{M}_{kj}$ with $j > k$ and as a negative example $(\boldsymbol{x}, 0)$ to all $\mathcal{M}_{ik}$ with $i < k$. Classifier $\mathcal{M}_{ij}$ is intended to discriminate instances with class label λ_i from those with class label λ_j.

Given the outputs of the base classifiers when presented with a test instance $\boldsymbol{x}$, a decision has to be made for the final classification of that instance. As mentioned in the introduction, we focus on *weighted voting* (WV) in order to make this decision.

The WV strategy considers the output $s_{ij} = \mathcal{M}_{ij}(\boldsymbol{x})$ as a weighted "vote" for class label λ_i and $s_{ji} = 1 - s_{ij}$ as a weighted vote for λ_j. Each class label λ_i is scored in terms of the sum of its votes

$$s_i = \sum_{1 \leq j \neq i \leq m} s_{ij} \quad , \tag{6.1}$$

and the class label with the maximal sum of votes is predicted.

6.2.2 Label Ranking Setting

The setting of label ranking can be seen as an extension of the conventional setting of classification (Har-Peled *et al.*, 2002). Each instance $\boldsymbol{x}$ is now associated with a total order of the class labels, and we write $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ to indicate that λ_i is preferred

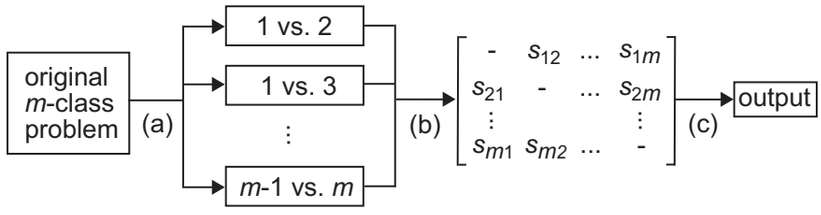


Figure 6.1: Basic structure of learning by pairwise comparison: (a) decomposition into binary classification problems, (b) base classifiers provide predictions, and (c) aggregation into one final prediction for the test instance.

to λ_j in the ranking associated with $\boldsymbol{x}$. The top-label, i.e., the winner class label, is said to be on position 1 in the ranking.

A ranking $\succ_{\boldsymbol{x}}$ can be identified with a permutation $\tau_{\boldsymbol{x}}$ of the set $\{1, \ldots, m\}$. The class of all permutations of this set is denoted by $\mathcal{S}_m$. For ease of presentation, it is convenient to define $\tau_{\boldsymbol{x}}$ such that $\tau_{\boldsymbol{x}}(i) = \tau_{\boldsymbol{x}}(\lambda_i)$ is the position of class label λ_i in the ranking. We further assume that there exists a probability distribution $\mathbb{P}(\cdot \,|\, \boldsymbol{x})$ for each $\boldsymbol{x} \in \mathcal{X}$ such that, for every $\tau \in \mathcal{S}_m$,

$$\mathbb{P}(\tau \,|\, \boldsymbol{x}) \tag{6.2}$$

is the probability that $\tau_{\boldsymbol{x}} = \tau$ (i.e., for each permutation there is a probability that it is the correct permutation for the instance under consideration).

Label ranking can be used for conventional classification purposes by associating the true class label $\lambda_{\boldsymbol{x}}$ with the top-label in the true ranking $\tau_{\boldsymbol{x}}$, i.e., $\lambda_{\boldsymbol{x}} = \tau_{\boldsymbol{x}}^{-1}(1)$. It follows that the probability of observing the class label $\lambda_{\boldsymbol{x}} = \lambda_i$ given $\boldsymbol{x}$ corresponds to the probability that λ_i occurs as the top-label in a ranking τ. This probability is computed by summing the probabilities of all possible rankings in which the label is at the first (top) position:

$$\mathbb{P}(\lambda_i \,|\, \boldsymbol{x}) = \sum_{\tau \in \mathcal{S}_m \,:\, \tau^{-1}(1) = i} \mathbb{P}(\tau \,|\, \boldsymbol{x}) \ .$$

Since the label ranking setting is relatively new in the machine learning field, we advise the non-familiar reader to read Section 2.4 again for a more detailed review.

6.2.3 Predicting a Label Ranking

Pairwise learning for classification can be extended to the setting of label ranking as follows (Fürnkranz and Hüllermeier, 2003). It is natural to interpret the output of base classifier $\mathcal{M}_{ij}$ as a soft decision whether $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ or $\lambda_j \succ_{\boldsymbol{x}} \lambda_i$. More specifically, we interpret the closer the output of $\mathcal{M}_{ij}$ to 1, the stronger the preference $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ is supported. Formally, we express a soft decision by a valued preference relation

$\mathcal{R}_{\boldsymbol{x}}$ for a test instance $\boldsymbol{x}$:

$$\mathcal{R}_{\boldsymbol{x}}(\lambda_i, \lambda_j) = \begin{cases} \mathcal{M}_{ij}(\boldsymbol{x}) & \text{if} \quad i < j \\ 1 - \mathcal{M}_{ij}(\boldsymbol{x}) & \text{if} \quad i > j \end{cases} \ . \tag{6.3}$$

Given a valued preference relation $\mathcal{R}_{\boldsymbol{x}}$ for an instance $\boldsymbol{x}$, the question is how to derive a label ranking from it. This question is non-trivial since such a relation does not always suggest a unique ranking in an unequivocal way (Fodor and Roubens, 1994; Hüllermeier and Brinker, 2008). Nonetheless, we remark that the weighted voting strategy can be extended to the prediction of a label ranking in a consistent and straightforward way. To this end, class labels are simply ordered according to their total scores, which are again interpreted as individual degrees of support. Thus, each class label λ_i is evaluated by means of the sum of its weighted votes

$$s_i = \sum_{1 \leq j \neq i \leq m} \mathcal{R}_{\boldsymbol{x}}(\lambda_i, \lambda_j) \ , \tag{6.4}$$

and a ranking is obtained by ordering according to these evaluations:

$$\lambda_i \succ_{\boldsymbol{x}} \lambda_j \Leftrightarrow s_i > s_j \ . \tag{6.5}$$

Like for weighted voting in the classification setting, possible ties can be broken at random or decided in favour of the majority class. In summary, the basic structure as represented in Fig. 6.1 remains identical after a reinterpretation of the s_{ij}, namely the scores represent a degree of support for a pairwise preference.

At first sight, the above prediction rule may appear rather ad-hoc (as is also the case with weighted voting in the classification setting). However, recently it has been shown by Hüllermeier *et al.* (2008) that in the label ranking setting, under mild technical assumptions, (6.5) is a risk minimiser with respect to the sum of squared rank differences as a loss function on rankings. In other words, the prediction rule (6.5) maximises the expected Spearman rank correlation between the predicted ranking and the true ranking.

6.2.4 Benefits of the Label Ranking Setting

We believe that analysing the framework of learning by pairwise comparison in the setting of label ranking is useful for several reasons. Each "vote" of a classifier has a clear and consistent semantics in this setting, namely the degree that λ_i should be ranked higher or lower than λ_j in $\tau_{\boldsymbol{x}}$. More specifically, from a probability perspective, we have the following intuitive interpretation:

$$s_{ij} = \mathbb{P}(\lambda_i \succ_{\boldsymbol{x}} \lambda_j) \ .$$

This offers an interesting alternative to the conventional classification setting in which an output s_{ij} is usually interpreted as the *conditional* probability of λ_i given that the class label is either λ_i or λ_j (Hastie and Tibshirani, 1998):

$$s_{ij} = \mathbb{P}(\, \lambda_{\boldsymbol{x}} = \lambda_i \,|\, \boldsymbol{x},\, \lambda_{\boldsymbol{x}} \in \{\lambda_i, \lambda_j\}\,) \ . \tag{6.6}$$

Without going into much detail, we mention three reasons why this interpretation is not uncritical, neither semantically nor technically.

- For a new test instance $\boldsymbol{x}$, which is submitted to all base classifiers, the interpretation (6.6) is actually only valid for those $\mathcal{M}_{ij}$ for which $\lambda_{\boldsymbol{x}} \in \{\lambda_i, \lambda_j\}$ since the probability is conditioned on that event (Cutzu, 2003). In other words, a learner $\mathcal{M}_{ij}$ with $\lambda_{\boldsymbol{x}} \notin \{\lambda_i, \lambda_j\}$ is not "competent" for $\boldsymbol{x}$ and, therefore, its prediction is meaningless. In label ranking, this problem does not exist since class label predictions are replaced by preferences and, by definition, each base classifier is competent for all instances (since for each instance there exists a total order of the class labels).

- Most machine learning techniques, even those that perform well for classification, actually yield poor probability estimates (Zadrozny and Elkan, 2002; Niculescu-Mizil and Caruana, 2005). This again calls the interpretation (6.6) into question. There is research dealing with transforming classifier's outputs into accurate estimations of true conditional probabilities. However, the results show that calibrating the outputs (scores) is classifier-dependent, sometimes related to some characteristics of the data sets, and difficult to evaluate in a consistent way; see Zhang and Yang (2004) and references therein for more details on this difficult task.

- Finally, the problem of deriving the probabilities $\mathbb{P}(\lambda_1 \,|\, \boldsymbol{x}), \ldots, \mathbb{P}(\lambda_m \,|\, \boldsymbol{x})$ from the pairwise probabilities (6.6) is non-trivial. It involves $m-1$ variables and $m(m-1)/2$ equality constraints, and thus the system does usually not have a unique solution. Different solution methods have different motivations and assumptions, and therefore it is difficult to predict which method is most suitable for the data set under consideration (Wu *et al.*, 2004).

By approaching the aggregation task within the setting of label ranking, we avoid the first problem. Adaptive voting, to be introduced in the next section, addresses the second problem. Namely, instead of requiring a classifier to output probabilities directly, we derive them indirectly by fitting a model to the classifier's outputs. Finally, the third problem also does not occur in the label ranking setting since the probability distribution over all rankings (6.2) can be used in a natural way to derive the posterior probabilities for all the class labels.

6.3 Adaptive Voting

In this section, we introduce adaptive voting as a novel aggregation strategy for pairwise learning. We present its formal framework (6.3.1), we show that it is optimal in the sense that it yields a MAP prediction under certain model assumptions (6.3.2), and we discuss the validity of these assumptions (6.3.3).

6.3.1 Formal Framework

Consider that, for a particular test instance $\boldsymbol{x}$, the predictions (scores) of all base classifiers are given by

$$s(\boldsymbol{x}) = \{s_{ij} = \mathcal{M}_{ij}(\boldsymbol{x}) \,|\, 1 \leq i \neq j \leq m\} \ . \tag{6.7}$$

Adopting a probabilistic perspective, we assume that the output s_{ij} is a random variable. Its distribution depends on whether $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ or $\lambda_j \succ_{\boldsymbol{x}} \lambda_i$; we shall denote the former event by E_{ij} and the latter by E_{ji}. For classifier $\mathcal{M}_{ij}$ to be accurate, it is natural that values of s_{ij} close to 1 are more probable than values close to 0 when E_{ij} occurs, and vice versa when E_{ji} occurs.

We can expect that different types of classifiers often yield different types of score distributions. So, in essence, any distribution (parametric or non-parametric) can be used to fit the scores, depending on what is appropriate given the base classifiers. Nonetheless, a reasonable assumption, also confirmed by empirical evidence, is that the outputs s_{ij} follow a monotone increasing (decreasing) function that has a more or less exponential shape when E_{ij} (E_{ji}) occurs. Hence, to model the scores, we use an exponential distribution which is truncated to $[0, 1]$. This assumption will also prove useful to show that weighted voting is a good approximation of our adaptive voting strategy. But again, we emphasize that any distribution can be used in AV and we give more details later on.

The truncated exponential distribution is given by

$$\mathbb{P}(s_{ij} \,|\, \tau) = c \cdot \exp\left(-\alpha_{ij} \cdot d(s_{ij})\right) \ , \tag{6.8}$$

where $c = \alpha_{ij}/(1 - \exp(-\alpha_{ij}))$ is a normalising constant and α_{ij} is a positive scalar. The term $d(s_{ij})$ is the prediction error defined as $1 - s_{ij}$ when E_{ij} occurs, and as s_{ij} when E_{ji} occurs. So, in other words, this term is the distance from the "ideal" outputs 1 and 0, respectively, depending on the event that occurred. Figure 6.2 gives a representative illustration where the exponential distributions are shown together with the empirical score histograms, which we obtained by applying a classifier.

We note that the higher the constant α_{ij} in (6.8), the more precise the outputs of the classifer $\mathcal{M}_{ij}$ are in the sense of stochastic dominance. For every constant $t \in (0, 1)$, the probability to make a small prediction error $d(s_{ij}) \leq t$ increases by increasing α_{ij}. Hence, adapting α_{ij} is a means to take the strength of $\mathcal{M}_{ij}$ into account. This can be done, for example, by using a maximum likelihood estimation, i.e., by maximising the log-likelihood function

$$\begin{aligned} LL(\alpha_{ij}) &= \sum_{k=1}^{n_{ij}} \log \mathbb{P}(s_{ij}^k \,|\, \tau_k) \\ &= n_{ij} \log(\alpha_{ij}) - n_{ij} \log(1 - \exp(-\alpha_{ij})) - \alpha_{ij} \sum_{k=1}^{n_{ij}} d_k \ , \end{aligned} \tag{6.9}$$

where n_{ij} is the number of examples used to train $\mathcal{M}_{ij}$, $s_{ij}^k = \mathcal{M}_{ij}(\boldsymbol{x}_k)$ is the prediction of the base classifier for the k-th validation instance $\boldsymbol{x}_k$, and $d_k \in [0, 1]$

is the corresponding prediction error. Setting the derivative of (6.9) with respect to variable α_{ij} equal to zero gives the following implicit solution:

$$\alpha_{ij} = \left(\bar{d} + \frac{1}{\exp(\alpha_{ij}) - 1} \right)^{-1} \quad , \tag{6.10}$$

where $\bar{d} = \sum_{k=1}^{n_{ij}} d_k / n_{ij}$ is the mean prediction error. This equation cannot be solved explicitly for α_{ij}, but it can be used as an iteration function; see Fig. 6.3 for a visualisation of this function. Thus, starting with an initial value, the value of α_{ij} is updated according to (6.10) and this is repeated until convergence. A good initial value is $1/\bar{d}$ since the second term in the sum of (6.10) goes fast to zero when alpha is increased. So, the initial value is already close to the solution, and indeed, the whole iteration converges extremely quickly.

For the sake of simplicity and for ease of exposition, we assume equal prior probabilities for the events E_{ij} and E_{ji}. Then, the following posterior probabilities $p_{ij} = \mathbb{P}(E_{ij} \,|\, s_{ij})$ and $p_{ji} = \mathbb{P}(E_{ji} \,|\, s_{ji})$ are obtained by applying Bayes' rule:

$$\begin{aligned} p_{ij} = \frac{\mathbb{P}(s_{ij} \,|\, E_{ij})}{\mathbb{P}(s_{ij})} &= \frac{1}{1 + \exp(\alpha_{ij}(1 - 2s_{ij}))} \\ p_{ji} = 1 - p_{ij} &= \frac{1}{1 + \exp(-\alpha_{ij}(1 - 2s_{ij}))} \end{aligned} \quad , \tag{6.11}$$

For strong classifiers with a large α_{ij}, these probabilities are "reinforcements" of the original scores s_{ij} in the sense that the p_{ij} are more toward the extreme values 0 and 1. In contrast, original scores are weakened for classifiers with small α_{ij}, i.e., the scores are pushed toward the indifference value of 0.5. Figure 6.4 gives an illustration of these effects. Making an idealized assumption of independence for the base classifiers, the probability $\mathbb{P}(\tau) = \mathbb{P}(\tau \,|\, \boldsymbol{x})$ of a ranking $\tau \in \mathcal{S}_m$, given the predictions (6.7) and corresponding posterior probabilities as defined above, is

$$\mathbb{P}(\tau) = c \cdot \prod_{i,j \in \{1,\ldots,m\} \,:\, \tau(i) < \tau(j)} p_{\tau^{-1}(i),\tau^{-1}(j)} \quad , \tag{6.12}$$

where c is a normalising constant. Essentially, this corresponds to a model known as Babington Smith in the statistical literature (Marden, 1995). The resulting probability distribution over $\mathcal{S}_m$ defined by (6.12) can serve as a point of departure for various types of inferences. In the following, we shall focus on classification, i.e., estimating the top-label in a ranking. Needless to say, as the number of different rankings $|\mathcal{S}_m| = m!$ can become very large, it is important to avoid an explicit construction of the distribution over $\mathcal{S}_m$.

6.3.2 MAP Classification

In the conventional classification setting, one is interested in the conditional probabilities $\mathbb{P}(\lambda_k \,|\, \boldsymbol{x})$, $\lambda_k \in \mathcal{L}$. Recalling that $\lambda_{\boldsymbol{x}} = \lambda_k$ means that λ_k is the top-label in

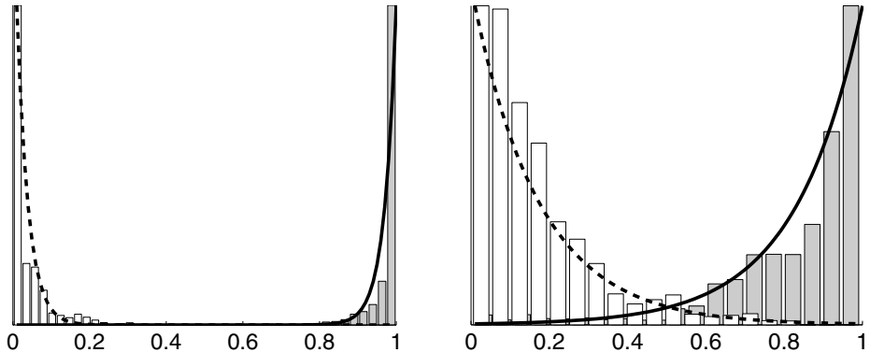


Figure 6.2: Two examples of an empirical distribution of s_{ij} (gray bars) and s_{ji} (white bars) together with the estimated truncated exponentials. The base classifier depicted in (a) is visibly more accurate and certain in its predictions than the classifier in (b). The height of the bars are scaled to match the distributions.

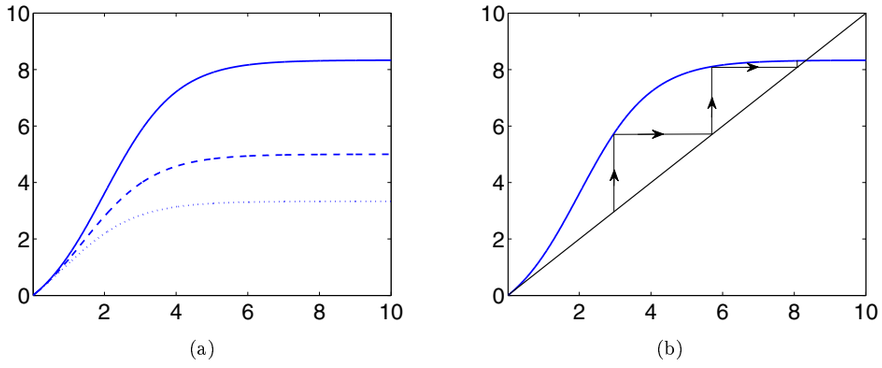


Figure 6.3: Illustrations for the maximum likelihood estimation: (a) the iteration procedure as a function of α_{ij} where the mean prediction error is 0.12 (solid curve), 0.2 (dashed curves), and 0.3 (dotted curve), and (b) example showing the quick convergence of the iteration.

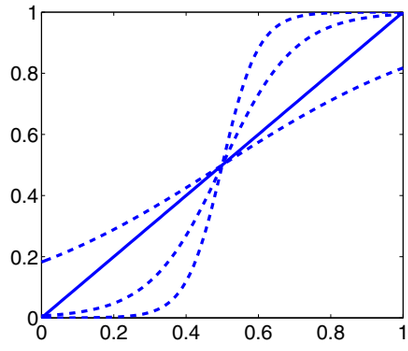


Figure 6.4: Original scores s_{ij} (solid curve) and transformations p_{ij} (dashed curves) for α_{ij}-values 1.5, 5, and 10. The higher the value of α_{ij}, the more we approach the step function, representing the strongest reinforcement possible.

ranking $\tau_{\boldsymbol{x}}$, we have the following:

$$\begin{aligned}
\mathbb{P}(\lambda_k \,|\, \boldsymbol{x}) &= c \cdot \sum_{\tau \in \mathcal{S}_m \,:\, \tau^{-1}(1)=k} \mathbb{P}(\tau \,|\, \boldsymbol{x}) \\
&\propto \prod_{1 \le k \ne i \le m} p_{ki} \underbrace{\sum_{\substack{\tau \in \mathcal{S}_m : \\ \tau^{-1}(1)=k}} \prod_{2 \le i < j \le m} p_{\tau^{-1}(i),\tau^{-1}(j)}}_{*=1} \\
&= \prod_{1 \le k \ne i \le m} p_{ki} \quad . \qquad (6.13)
\end{aligned}$$

Lemma 6.1. *The above expression (*) evaluates to 1 since it has the form* $\sum_{\boldsymbol{\ell} \in \{0,1\}^h} \prod_{i=1}^{h} u_i^{(\ell_i)}$ *with* $u_i^{(0)} = u_i$, $u_i^{(1)} = 1 - u_i$, *and by definition* $u_i \in [0,1]$.

Proof. We use ℓ_i to denote whether the i-th probability we are considering in the product, henceforth abbreviated by u_i, is the probability by itself ($\ell_i = 0$) or its complement ($\ell_i = 1$). In other words, it allows us to switch between the posterior probabilities as defined in (6.11). The proof we consider is by induction.

- **Base case:** $h = 1$ (implying a total of 3 classes).
  The product term only involves one element and taking the sum into account gives that $u_1^{(0)} + u_1^{(1)} = u_1 + (1 - u_1) = 1$.

- **Induction step:** $h \ge 2$ (implying more than 3 classes).
  Remembering that we are interested in the product of a fixed number of probabilities, summed over all possible configurations denoted by ℓ_i, $1 \le i \le h$,

we can derive the following:

$$\begin{aligned}\sum_{\boldsymbol{\ell}\in\{0,1\}^h}\prod_{i=1}^{h}u_i^{(\ell_i)} &= \sum_{l_h\in\{0,1\}}\sum_{\boldsymbol{\ell}\in\{0,1\}^{h-1}}\prod_{i=1}^{h}u_i^{(\ell_i)}\\ &= \sum_{l_h\in\{0,1\}}u_h^{(l_h)}\sum_{\boldsymbol{\ell}\in\{0,1\}^{h-1}}\prod_{i=1}^{h-1}u_i^{(\ell_i)}\\ &= \sum_{l_h\in\{0,1\}}u_h^{(l_h)}\end{aligned}$$

where the last step is obtained by applying the induction principle and the last sum is again 1 since we are considering a probability and its complement.

□

In the remainder of the chapter, we will score the class labels λ_i in terms of the logarithm of their probability and not the probability itself. The reason is that the logarithm is a monotone function of its arguments (and thus, it preserves the ordering) and, when considering probabilities, it is from a computational precision point of view better to compute with logarithms. Therefore, we define

$$w_i = \log\left(\mathbb{P}(\lambda_i \,|\, \boldsymbol{x})\right) = \sum_{1\leq j\neq i\leq m}\log(p_{ij}) = \sum_{1\leq j\neq i\leq m} w_{ij} \ , \tag{6.14}$$

where the individual *adapted scores* are defined as

$$w_{ij} = -\log(1+\exp(\alpha_{ij}(1-2s_{ij}))) \ . \tag{6.15}$$

Hence, the MAP prediction

$$\lambda_{\boldsymbol{x}}^{MAP} \ = \ \arg\max_{i=1\ldots m}\log\left(\mathbb{P}(\lambda_i \,|\, \boldsymbol{x})\right) \tag{6.16}$$

is obtained by finding the class label λ_i for which the sum (6.14) is maximal. We call this aggregation strategy *adaptive voting* (AV) because the original scores s_{ij} are replaced by adapted scores w_{ij} incorporating the strength of the base classifiers $\mathcal{M}_{ij}$. In this way, base classifiers with high performance are seen as more reliable than classifiers with lower performance, and the aggregation strategy becomes less sensitive to (likely incorrect) outputs from the weak and unreliable classifiers.

6.3.3 Discussion of Model Assumptions

Even though is is clear that, without any model assumptions, it is impossible to justify a predictor in a theoretical way, let alone to prove its optimality, it is legitimate to ask whether our concrete assumptions are both reasonable and realistic. Therefore, we briefly comment on the two main assumptions underlying our adaptive voting method.

The assumption of exponentially distributed scores essentially comes down to assuming a monotone behaviour, expressing that correct scores are more probable than incorrect ones. This assumption is clearly reasonable for better-than-random classifiers. Moreover, since we actually use a truncated distribution restricted to the unit interval, the model can capture almost all reasonable shapes, ranging from the uniform to the extreme boundary distributions. Practically, as will be shown later in our experimental studies, the assumption does indeed hold true for classifiers such as multilayer perceptrons, while being violated by others such as decision trees. It is clear, however, that no distribution will be able to agree with all type of classifiers. Besides, an accurate modelling of distributions such as those produced by decision trees seems to be extremely difficult.[3]

The second main assumption is the independence of predictions produced by the base classifiers. Even though it is unlikely to be completely valid in practice, we mention four reasons to defend it. First, independence is routinely assumed in statistics, mainly because without this simplifying assumption, a formal analysis is greatly complicated and often not possible at all. Second, special types of independence assumptions are also made by other successful machine learning methods, notably the well-known Naive Bayes classifier. Third, one may argue that assuming any specific type of dependency between the classifiers is at least as speculative as assuming independence. Fourth, we note that the assumption of independence is implicitly made by weighted voting. In this sense, it is even a necessary prerequisite for our goal to establish a formal connection between adaptive voting and weighted voting (as we shall do in Subsection 6.5.1).

6.4 Simulation Studies with Synthetic Data

In this section, we present two simulation studies to investigate the effectiveness of AV and its robustness toward estimation errors when fitting the truncated exponential distributions (6.4.1). We compare the classification performance with WV since this aggregation strategy has shown good performance in practice (6.4.2).

6.4.1 Experimental Setup and Results

The setup of the simulation studies is as follows. We assume that the scores s_{ij} are generated independently according to the truncated exponential distribution (6.8). For a fixed number of classes m, we generate a random set of α_{ij}-values ($1 \leq i < j \leq m$), each one in the range $[1, 3]$, and then proceed as follows. A random ranking is generated and the output of the base classifiers are computed according to their distributions. These outputs are directly used by WV in order to make a final classification, while AV first adapts them to take the strength of the classifiers into account. The predictions of the aggregation strategies are compared with the

[3]We tried a number of more flexible models such as Beta distributions. However, these distributions often involve additional complications and sometimes produce undesirable effects such as non-monotone functions. The best fits that we could obtain with such distributions did not improve the experimental results.

top-label in the ranking. This process is repeated for 300.000 times, each time with different rankings and scores. The test statistic to our interest is the expected *relative* improvement $(a - b)/b$, where a is the classification rate of AV and b that of WV.

In Fig. 6.5, we show the expected relative improvement as a function of the strength of the base classifiers. More specifically, each position $t \in [1, 15]$ on the horizontal axis shows the result when the initially generated α_{ij}-values are multiplied with t. The plots are representative for all the initial values that we have generated; in total we generated 200 curves and at least 95% of these curves lie in-between the plotted 20 curves. We may formulate the following three observations.

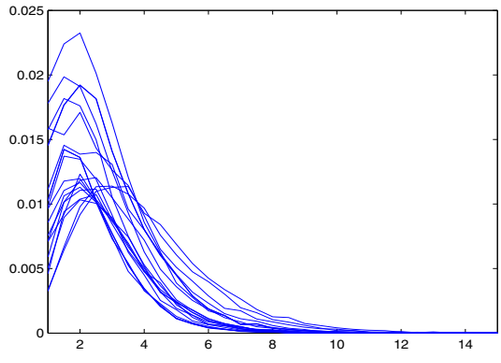
First, AV improves on WV when its model assumptions are met. Second, there is hardly any difference between the strategies for classifiers with large α_{ij} (rightmost part of the curves). The reason is that, for very strong base classifiers, the true class label is likely to be a clear winner among all class labels, so an extra adaptation of the scores becomes unnecessary. Third, the benefit of AV increases with the number of classes since there are more scores to adapt, and hereby, more possibilities to correct the classification of WV. We clearly see this trend when we consider the absolute values on the vertical axis and the variance among the curves (variances are not shown in order to keep the figures easily readable, but they are decreasing with increasing number of class labels).

The second simulation study investigates the robustness of AV toward errors in estimating the α_{ij}. The outputs of the base classifiers are again computed with the randomly generated α_{ij}, but AV uses a noisy version of them in its calculations. Noise is incorporated by replacing an α_{ij}-value by a randomly drawn number that at most deviates respectively 10%, 20%, and 30% from the true value. Figure 6.6 shows the expected relative improvement for each of these three noise levels. The depicted solid curves are again a representative set of all 200 curves that we generated. The curve corresponding to the results obtained by using the true α_{ij} is not depicted since it is visually almost indistinguishable from the other curves. In general, the true curve can be considered as the highest curve in the figure. For a similar reason, we do not present results for $m = 10$. We may formulate the following three observations from the results.

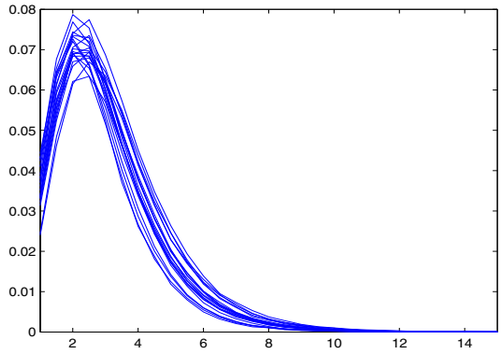
First, in general, AV is remarkably robust toward estimations errors in the α_{ij}. Second, the robustness increases with the number of classes. Third, independent of the number of classes, inspecting differences in terms of absolute numbers indicates that adding noise has most impact for $\alpha_{ij} < 7$ (these differences represent more than 95% of all differences). This corresponds with our intuition.

6.4.2 Discussion of the Experimental Results

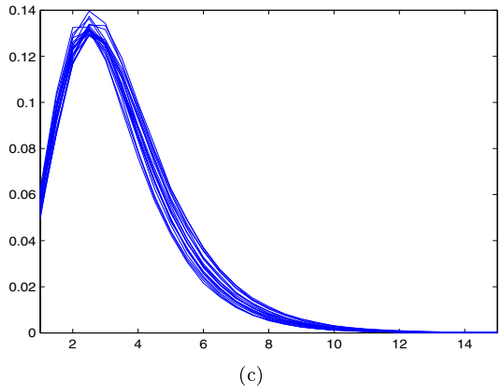
In the above experiments, it is not unintentional that the expected improvement is computed with respect to weighted voting, which is known to perform extremely well in practice; see for example Hüllermeier and Fürnkranz (2004; 2009). Moreover, our experiments with real data sets and classifiers (to be detailed in Section 6.6) have shown that the predictions of AV and WV are quite likely to coincide, often with a probability significantly higher than 0.9.



(a)



(b)



(c)

Figure 6.5: Expected relative improvement in classification rate comparing AV and WV for: (a) $m = 3$, (b) $m = 6$, and (c) $m = 10$. Relative improvements are shown as a function of the strength of the base classifiers.

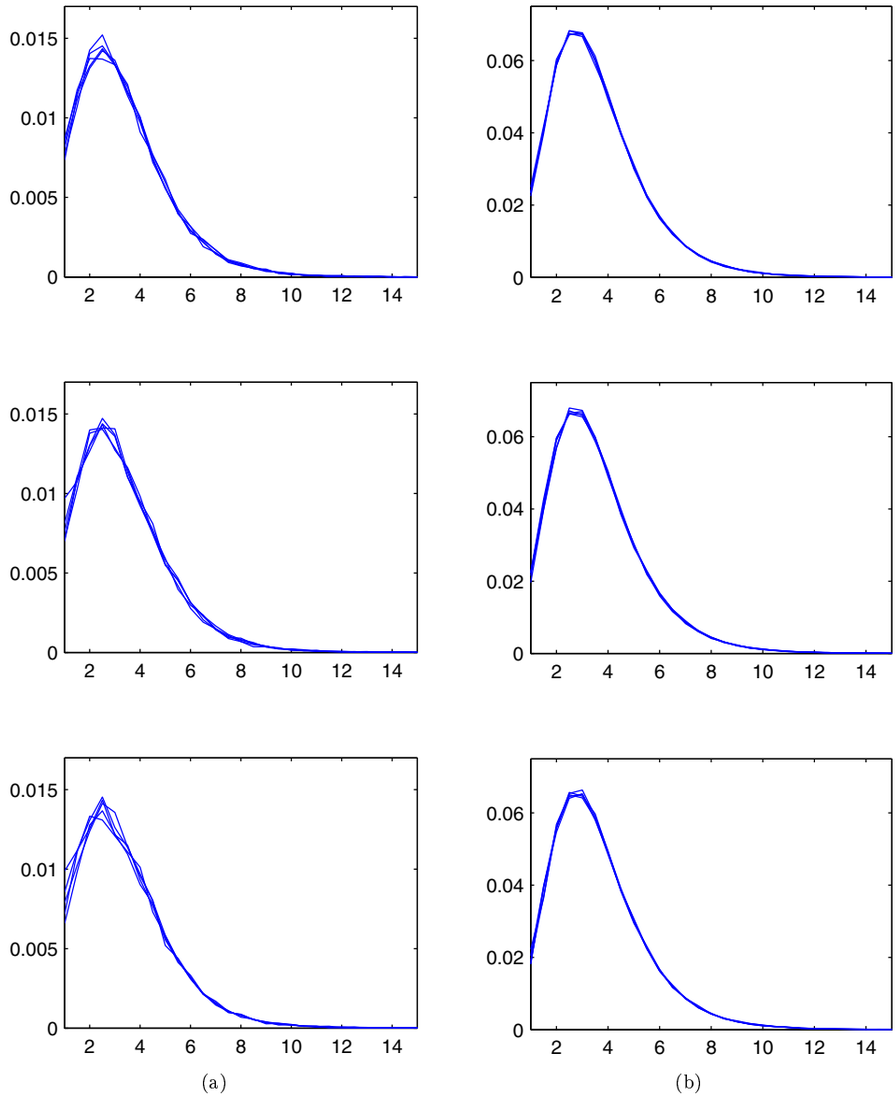


Figure 6.6: Expected relative improvement in classification rate comparing AV and WV for: (a) $m = 3$, (b) $m = 6$ when noise is added to the alphas (top row: 10%, middle row: 20%, and bottom row: 30%). Relative improvements are shown as a function of the strength of the base classifiers.

To explain this observation, we note that a good probability estimation is a sufficient but not necessary prerequisite for a good classification. More specifically, classification is quite robust toward inaccurate probability estimates since the classification remains correct as long as the highest estimated probability is assigned to the true class label. Consequently, AV and WV will coincide as long as the label with highest w_i does also receive the highest s_i. In fact, there is often a relatively clear winner among the candidate classes, especially if the underlying classification problem is simple or the base classifiers are strong (or both). Note that, as a consequence, all "reasonable" aggregation strategies will perform more or less the same in such cases, so that large differences in performance cannot be expected.

In the next section, the above line of reasoning will be substantiated more formally by showing that WV indeed provides a good approximation of AV.

6.5 Weighted Voting

We now focus on the weighted voting strategy, which has shown excellent performance in practice. A new and formal explanation of this observation is provided by proving that WV yields an approximation to the optimal AV prediction (6.5.1). In addition, we argue that WV can sometimes be considered as being more robust than AV, which is potentially advantageous from a classification point of view (6.5.2).

6.5.1 Approximate MAP Prediction

In this section, it will be shown that weighted voting can be seen as an approximation to adaptive voting, i.e., to the MAP prediction (6.16). Recalling the posterior probabilities (6.11) we derive from their logarithm

$$\begin{aligned} w_{ij} &= \log(\mathbb{P}(s_{ij} \mid E_{ij})) - \log(\mathbb{P}(s_{ij} \mid E_{ij}) + \mathbb{P}(s_{ij} \mid E_{ji})) \\ &= \alpha_{ij} \cdot s_{ij} - \alpha_{ij} + e(d_{ij}, \alpha_{ij}) \ , \end{aligned} \tag{6.17}$$

where

$$e(d_{ij}, \alpha_{ij}) = -\log\left(\exp(-\alpha_{ij} d_{ij}) + \exp(-\alpha_{ij}(1 - d_{ij}))\right) \ .$$

This term, which depends on α_{ij} and the prediction error $d_{ij} = d(s_{ij})$ of classifier $\mathcal{M}_{ij}$, is bounded in size by $|\alpha_{ij}/2 - \log(2)|$. Since $\exp(x) \approx 1 + x$ for small x, we have that $e(d_{ij}, \alpha_{ij})$ is small and nearly constant for small α_{ij}. For larger α_{ij}, we have $e(d_{ij}, \alpha_{ij}) \approx \log(1 + e^{-\alpha_{ij}}) \approx 0$ at least for d_{ij} close to 1 or close to 0, so that

$$w_{ij} \approx \alpha_{ij} \cdot (s_{ij} - 1) \ . \tag{6.18}$$

In summary, we may conclude that

$$w_i \approx \sum_{1 \le j \ne i \le m} \alpha_{ij} \cdot s_{ij} - \sum_{1 \le j \ne i \le m} \alpha_{ij}$$

when the α_{ij} are either not too large or when the predictions are precise, which in turn is likely in case of large α_{ij}. We note that this is in perfect agreement with the

results of our simulation studies in which the maximal differences between AV and WV have been observed for alphas of medium size.

If we furthermore assume that the strengths of the classifiers are not too different, that is, $\alpha_{ij} \approx \alpha$ for all $1 \leq i < j \leq m$, then we have

$$w_i \approx \alpha \sum_{1 \leq j \neq i \leq m} s_{ij} + \text{const} = \alpha \cdot s_i + \text{const} \quad . \tag{6.19}$$

In other words, the scores obtained by weighted voting yield an approximate affine transformation of the theoretically optimal score $\log(\mathbb{P}(\lambda_i))$ and, thus, are likely to produce the same or a similar ordering of the class labels λ_i, $i = 1, \ldots, m$. We may conclude that, under the above assumptions, weighted voting provides a good approximation of the MAP prediction (6.16).

Nevertheless, the above derivation has also shown that AV and WV may not coincide in cases where the α_{ij} are rather different and, moreover, when some base classifiers produce poor estimates. Thus, it is still possible that AV will produce better results in practice, and in fact, this hope is supported by our simulation results presented in the previous section. However, recalling that these results have been obtained under idealised conditions in which the model assumptions underlying AV are completely valid, one may also suspect that AV could fail if these assumptions are not satisfied. And indeed, apart from its approximation properties, WV seems to have the advantage of making less assumptions and, therefore, being more robust toward deviations from expected score distributions. Before presenting experimental results in the next section, we elaborate on this aspect in more detail.

6.5.2 Robustness Toward Inaccurate Scores

The adapted scores (6.15) are rescalings of the original scores $s_{ij} \in [0, 1]$ to a potentially larger range $[-\log(1 + \exp(\alpha_{ij})), -\log(1 + \exp(-\alpha_{ij}))] \subset (-\infty, 0]$. It follows that the sum of these adapted scores, w_i, may have a considerably higher variance than the corresponding s_i. Moreover, we note that a small score s_{ij} produced by a reasonable to very strong classifier leads to a large negative value for the adapted score, which in turn, leads to a large negative value for w_i.

So, while being correct and fully legitimate from a theoretical point of view, the adaptations in AV may become problematic if its model assumptions are violated since, in this case, the true class label may be *penalised* too severely and therefore by far loses the adaptive voting procedure.[4] The most extreme situation is when all pairwise scores are in favour of the true class label, except for one that has value 0. Then, the true class label cannot be predicted due to this single inaccurate score (the class label has score minus infinity, or equivalently, it is penalised by an infinite amount). Thus, it may happen that a few or even a single w_{ij} dominate the whole

[4] Formally, we consider the penalty of a class label λ_i as $-w_i$. Thus, according to (6.15), a class label λ_i is penalised when s_{ij} is small, and the degree of penalisation is in direct correspondence with the probability to observe the output s_{ij} given that λ_i is the true class label: The smaller s_{ij}, the smaller the probability and, hence, the higher the penalty $-w_i$.

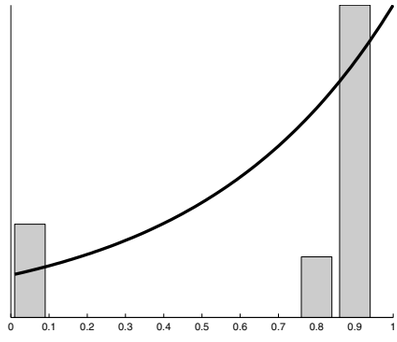


Figure 6.7: A score distribution with three possible values (0.05, 0.8, and 0.9) for $\lambda_i \succ \lambda_j$. The corresponding probabilities are respectively 0.20, 0.13, and 0.67. The maximum likelihood fit gives a value of $\alpha_{ij} \approx 1.3947$.

voting procedure, which is clearly a disadvantage since we cannot ensure that the domination is in favour of the true class label.

As an illustration, inspired by our experiments with decision trees (presented in the next section), we consider the fit of the distribution shown in Fig. 6.7. Since this distribution is discrete and far from exponential (only three scores can be produced), the maximum likelihood fit is obviously poor. More importantly, we note that the probability of the smallest score 0.05 is strongly underestimated. When this score is output (which happens with probability 0.2), the class label will be strongly punished and is unlikely to win the adapted voting procedure, even though it may be the true class label for the instance under consideration. More concretely, to elaborate more on this important issue, consider a scenario with $m = 4$ classes and assume that all base classifiers have the distribution shown in Fig. 6.7. Moreover, consider for a moment that λ_1 is the true class label, and that the following scores are produced:

$$[s_{ij}]_{i \neq j} = \begin{bmatrix} - & 0.90 & 0.90 & 0.05 \\ 0.10 & - & 0.90 & 0.80 \\ 0.10 & 0.10 & - & 0.80 \\ 0.95 & 0.20 & 0.20 & - \end{bmatrix} .$$

Obviously, the learner $\mathcal{M}_{14}$ has made an incorrect prediction. WV tolerates this error in the sense that it still assigns the highest score (namely 1.85) to λ_1. According to AV, however, class label λ_2 is better than λ_1: the latter is penalised by ≈ -2.07, while the former has a smaller penalty of ≈ -2.04. As explained above, the main reason is that the small score of 0.05 produced by $\mathcal{M}_{14}$ is over-penalised.

We can see the above effect also from an opposite perspective, i.e., the original scores s_{ij} used in weighted voting can be considered as "regularised" scores which are normalised to the range $[0, 1]$, thereby especially reducing the effect of small scores. While perhaps being suboptimal for probability estimation, this can be reasonable

from a classification point of view because, as long as the true class label receives correct votes (i.e., high scores), nothing will change anyway. However, if the true class label receives one or more incorrect votes, an aggregation strategy which is tolerant toward low scores is more likely to preserve this class label as the overall winner than a strategy which is more sensitive in this regard.

Interestingly, the situation is to some extent comparable to estimating conditional probabilities of attributes given class labels, $\mathbb{P}(a \mid \lambda_i)$, in Naive Bayes classification. Estimating these probabilities by relative frequencies yields unbiased estimates, but it causes the problem that small probabilities have an extreme influence on the rank position of a class label. In particular, if a single probability is 0 (since the attribute value of a has not yet been observed for λ_i), then multiplication of all conditional probabilities causes the probability of the class label to become 0 as well. In practice, probabilities are therefore estimated by using a Laplace correction or another smoothing technique (Domingos and Pazzani, 1997). Clearly, a similar problem occurs in AV so that, as soon as one of the probabilities p_{ki} in (6.13) becomes small, the probability of the true class label λ_k becomes small as well. Correcting transformed scores in AV is however not a trivial task since scores lie in different intervals (unbounded from one side).

6.6 Experimental Analysis

In this section, we provide an extensive empirical evaluation and comparison between WV and AV using benchmark data sets. We explain the choice of data sets and classifiers (6.6.1), the experimental setup (6.6.2), and we analyse the results (6.6.3).

Our experiments also include a third aggregation strategy, called binary voting, for two reasons. First, it is often used as a simple alternative to WV. Second, this strategy maps the original scores s_{ij} to transformations $b_{ij} \in \{0, 1\}$ where $b_{ij} = 1$ iff $s_{ij} \geq 0.5$. It follows that binary voting (BV) can also be seen as a reinforcement of the outputs of the base classifiers, although this happens independently of their strength (in Fig. 6.4, this reinforcement would correspond to the step function).

6.6.1 Data Sets and Classifiers

To compare BV, WV, and AV, experiments have been conducted on a collection of 17 benchmark data sets obtained from the UCI machine learning repository and the StatLib project (Asuncion and Newman, 2007; Vlachos, 2008). We have chosen these data sets because they vary greatly in size, number of classes, and other characteristics such as class distribution; see Table 6.1. We used several learning algorithms to produce base classifiers. Below we present three of them. All learning algorithms produce multi-class classifiers, but it has been shown that even these classifiers benefit from a pairwise decomposition (Fürnkranz, 2003). In addition, and more importantly, we have chosen these algorithms because they are representative in the sense that they learn base classifiers that satisfy or do not satisfy to a certain degree the AV assumptions.

Table 6.1: The seventeen data sets, where the column headings refer to: (1) reference number, (2) data set name, (3) number of instances, (4) number of classes, (5) percentage of the minority class, and (6) percentage of the majority class.

#	name	size	m	% min class	% maj class
1	analcatdata-authorship	841	4	6.54	37.69
2	balance-scale	625	3	7.84	46.08
3	cars	1728	4	3.76	62.56
4	cmc	1473	3	22.61	42.70
5	eucalyptus	736	5	14.27	29.08
6	glass	214	6	4.21	35.51
7	mfeat-fourier	2000	10	10.00	10.00
8	mfeat-karhunen	2000	10	10.00	10.00
9	mfeat-morphological	2000	10	10.00	10.00
10	mfeat-zernike	2000	10	10.00	10.00
11	optdigits	5620	10	9.86	10.18
12	pendigits	10992	10	9.60	10.41
13	page-blocks	5473	5	0.55	89.77
14	segment	2310	7	14.29	14.29
15	vehicle	846	4	23.52	27.77
16	vowel	990	11	9.09	9.09
17	waveform	5000	3	33.06	33.84

Our first classifier is a multilayer perceptron (MLP) where each network node has a sigmoid transfer function. As we could verify by statistical goodness-of-fit tests, the scores produced by MLPs are sufficiently well in agreement with our model assumptions. That is, the scores s_{ij} produced by a classifier $\mathcal{M}_{ij}$ can usually be fitted by a truncated exponential model (6.8); see again Fig. 6.2. It is our claim that AV can outperform WV and BV for this classifier. We make the same claim for our second classifier, which is a distance-weighted k-nearest neighbour classifier (k-NN), although admittedly, the exponential model is this time less clearly pronounced. The classifier computes scores

$$s_{ij} = \frac{\sum_{l=1}^{k} e_l/d_l}{\sum_{l=1}^{k} 1/d_l} \ ,$$

where d_l is the distance from the test instance to its l-th neighbour, $e_l = 1$ if this neighbour is from class λ_i and $e_l = 0$ if the class is λ_j. Just for comparison, and also to investigate the robustness of adaptive voting toward strong violations of its model assumptions, we included a third base classifier: J48, an implementation of the C4.5 decision tree learner. This classifier outputs relative class frequencies in the leaves as scores. Such scores often do not exhaust the whole interval $[0, 1]$ but instead only produce a limited number of distinct values. This makes a good estimation of the α_{ij} difficult; see Fig. 6.8. Therefore, we believe that AV will not gain in performance.

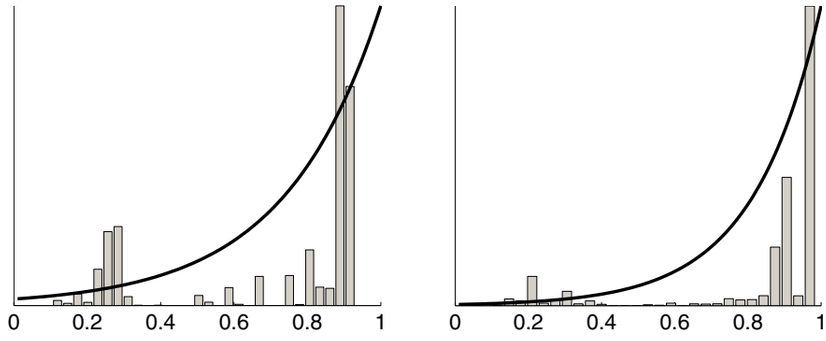


Figure 6.8: Two examples of an empirical distribution of the scores s_{ij} for $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ together with the estimated truncated exponential distribution when using J48 as base classifier. The height of the bars are scaled to match the distributions.

6.6.2 Experimental Setup

In our experiments, we used the `WEKA` machine learning software (Witten and Frank, 2005). The experimental setup was as follows. For all three base learners we used the default options, except for a variable learning rate in MLP, a fixed number of nearest neighbours in k-NN (namely $k = 10$), and Laplace correction for probability estimation in J48. Stratified ten-fold cross validation is applied and repeated for five times, each time with a different random permutation of the data set. The alpha values for AV are obtained by maximum likelihood estimation on an independent 20% subset of the training data. Ties among the ranking of class labels occurred for BV, but only sporadic on some data sets. We simply resolved these ties at random.

6.6.3 Experimental Results

In a first experiment, we compute error rates as averages over the cross-validation runs. For readability, we refrain from presenting the error rates for all aggregation strategies and base classifiers; all results can be found in Hüllermeier and Vanderlooy (2009a). We summarised these values as pairwise win-loss-equal statistics and with critical distance (CD) diagrams. Since both evaluation methods agreed on the conclusions that we can draw from them, we will only present the CD diagrams since they are easier to understand; see Fig. 6.9. The diagrams depict the result of the Nemenyi test which has been advocated as stronger than other widely-used significance tests (Demšar, 2006). The test compares the average ranks of the strategies over all data sets. A lower average rank implies a better aggregation strategy (thus, strategies on the right side in the diagrams are better) and strategies that are not significantly different when compared to each other are connected through a bold line. The significance level is $\alpha = 0.05$ which implies that two strategies are significantly different when the difference between their average ranks is at least $\mathrm{CD} = 0.88$.

The results confirm what could be expected from our theoretical considerations and the simulation results. More specifically, for MLPs we have a significant improvement by AV when compared to WV. Also, from the fact that BV is ranked better than WV, we may conclude that reinforcements are beneficial, yet the strengths of the base classifiers have to be taken into account. For the nearest neighbour classifier we have identical observations and conclusions although we cannot guarantee that the differences are statistically significant. For J48, the classifier that strongly violates our model assumptions, we see that WV wins significantly from BV, but not from AV which this time is ranked second. As a side note, we also tried to use unpruned trees with the idea that increasing the number of scores (making the score distribution less discrete) could give rise to a better fit of the exponential model. However, from statistical tests, we did not see a significant gain.

As a final experiment, we are interested in the robustness of the strategies with respect to inaccurate scores of the base classifiers. To investigate this issue, we note that the classifications of the aggregation strategies often coincide with a very high probability (Hüllermeier and Vanderlooy, 2009a). This implies that the strategies often make mistakes for the same instances, and a measure for how severe the mistake is will give a good indication about the robustness toward inaccurate votes. For this reason, we apply experiments in the same setting as above but replace the error rate as a loss function by the normalised position error which is defined by

$$\frac{\tau^{-1}(\lambda_x) - 1}{m-1} \in \{0, 1/(m-1), \ldots, 1\} \quad ,$$

where we recall that $\tau^{-1}(\lambda_x)$ is the position of the true class label $\lambda_{\boldsymbol{x}}$ in the predicted ranking τ and m the number of classes (Hüllermeier and Fürnkranz, 2007). Hence, the larger the normalised position error, the further away the true class label is from the top position in the predicted ranking, thus the more severe the incorrect prediction. The CD diagrams are presented in Fig. 6.10.

Interestingly, the ranking of the aggregation strategies is indeed sometimes different compared to that when we consider the error rate. For MLP we still have the same ranking and for k-NN we have that AV changed second place with BV, but differences are not significant. The ranking for J48 shows that AV makes the largest incorrect predictions in the sense that, when an incorrect prediction is made, the true class label is not at all among the labels that try to compete for the top label in the predicted ranking. So, large variance among the scores in combination with inaccurate votes is disadvantageous for adaptive voting. Weighted voting and binary voting are clearly more robust. All these results are again in correspondence with our previous theoretical analysis, simulation studies, and discussions.

In light of the above results, we may also explain why the binary voting strategy performs quite reasonable in terms of error rate and normalised position error, despite its apparent simplicity. We consider BV as a combination of AV and WV in the following sense. Like AV, it reinforces scores s_{ij}, albeit in a fixed rather than adaptive manner (always mapping to 0 or 1). It considers all classifiers as perfect, an assumption which is only approximately true for an ensemble of strong learners.

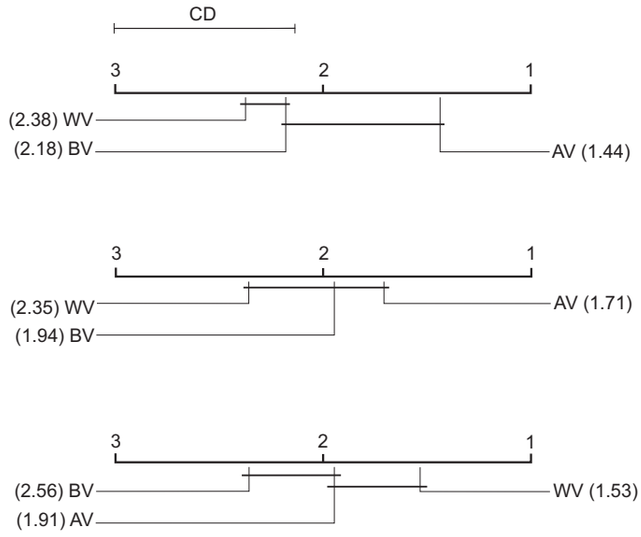


Figure 6.9: Comparison of aggregation strategies on the basis of the Nemenyi test, using the error rate: (top) MLP, (middle) k-NN, and (below) J48.

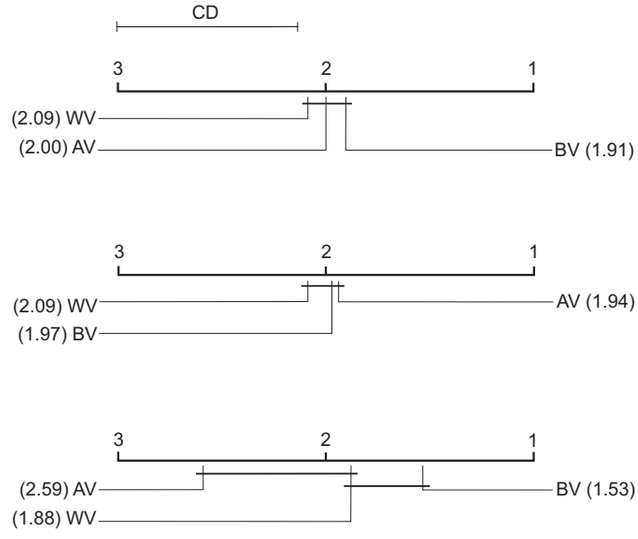


Figure 6.10: Comparison of aggregation strategies on the basis of the Nemenyi test, using the normalised position error: (top) MLP, (middle) k-NN, and (below) J48.

Like WV, however, the scores remain in the unit interval and thereby BV is less sensitive toward inaccurate probability estimates than AV.

6.7 Chapter Conclusions

In this chapter, we focussed on RQ 4: *Can we develop an aggregation strategy that is feasible in practice and shown to be optimal under reasonable conditions?* In this regard, we restricted ourselves to aggregation strategies that can be used in the learning by pairwise comparison framework, since this framework is well-suited for a formal analysis and it is most often used in practice.

To answer our RQ 4, we presented two important theoretically well-founded contributions. First, the strategy of adaptive voting has been derived in the formal setting of label ranking. Adaptive voting is a generalised voting strategy in which the predictions of base classifiers are adapted according to their strength. Under our model assumptions, we may conclude that it is provably optimal in the sense of yielding a MAP prediction of the class label of a test instance. Second, we offered hitherto missing theoretical arguments in favour of weighted voting as a quasi-optimal aggregation strategy in pairwise classification and, thereby, improve the understanding of its good performance in practice. Roughly speaking, we may conclude that the simple weighted voting approximates the optimal adaptive voting prediction. Moreover, compared to adaptive voting, weighted voting has the additional advantage of being more robust in situations where the AV model assumptions are violated.

Our empirical results are in perfect agreement with all theoretical considerations and discussions. In summary, we have shown that weighted voting is quite competitive, even though slight but consistent improvements can be achieved by adaptive voting provided its underlying model assumptions are approximately valid.

Chapter 7

Thesis Conclusion

The research presented in this thesis was motivated by the need for approaches that result in a safer and more reliable use of machine learning in order to allow for intelligence led policing. In this chapter, we provide a conclusive answer to the problem statement and the accompanying research questions. We start in Section 7.1 by restating and answering our four research questions. Then, we combine in Section 7.2 these answers to form a conclusive answer to the problem statement. Finally, we end with several directions for future work in Section 7.3.

7.1 Answer to the Research Questions

In the first chapter of the thesis, we presented three different but promising research directions and formulated four accompanying research questions. This section provides an answer to each of these questions. Since we have addressed each research question in a separate chapter, we refer to the chapter conclusions for more details.

7.1.1 Research Question 1

The first research question, that we have addressed in Chapter 3, deals with ranking instances from most likely positive to most likely negative. It reads as follows.

> **Research question 1**: To what extent is the AUC an effective performance metric for bipartite ranking when compared to its variants that consider the absolute values of the scores?

The question was motivated by the recent conjecture of several authors that enhancements of the AUC should be considered from a model evaluation and selection point of view, and not the AUC itself. These variants of the AUC consider small score margins as less reliable than large score margins, and therefore, the small score margins have less influence on the ranking performance value. However, from our theoretical and empirical results, we may conclude that none of the variants are as

effective as the AUC itself. The same holds true for any other variant that fits into our unifying framework of AUC estimates.

7.1.2 Research Question 2

Given that the AUC is a good ranking performance metric, it is interesting to ask how to learn a model that maximises the AUC. For the field of law enforcement, it is important that the predictions of the classifier can be understood, analysed, and interpreted. A decision tree allows for these important requirements. Therefore, the second research question was formulated as follows.

> **Research question 2**: How can the AUC of an interpretable and comprehensible model, such as decision trees, be optimised?

We addressed the second research question in Chapter 4, where we presented an extensive set of experiments and a formal analysis. From all our results, we may conclude that the AUC of decision trees can be improved by increasing the number of distinct scores, provided that the scores do not deviate too much from the true conditional class probabilities. Based on this finding, we tested with a simple but very effective approach to AUC-optimising decision trees by perturbing the feature vector and propagating an instance several times through the tree. It is important to note that our results generalise to other classifiers.

7.1.3 Research Question 3

Despite the benefits of ranking instances, the first research direction, it is still unknown *a priori* how many and which mistakes the classifier makes. For this reason we also pursued a second research direction, namely that of designing classifiers in such a way that they guarantee a preset classification performance. Thus, the third research question reads as follows.

> **Research question 3**: Can we develop a feasible approach by which a classifier is constructed that guarantees a preset classification performance on each class?

We provided an affirmative answer to this research question in Chapter 5 by means of introducing the ROC isometrics approach. The approach overcomes, or at least strongly alleviates, all three domain-specific problems that we discussed in the first chapter of this thesis (i.e., high error costs, disproportional privacy violations, and evolving class and cost distributions). We provided a formal and empirical analysis of the effectiveness and efficiency of the approach. We may conclude that the resulting reliable classifiers can be safely applied in practice, and therefore, the ROC isometrics approach is a valuable tool for law enforcement. In fact, it has already been successfully tested for credit card fraud detection (Vanderlooy *et al.*, 2006a).

7.1.4 Research Question 4

Our final research direction deals with multi-class problems. A common strategy to solve a multi-class problem is to decompose it into a series of binary problems. The question is then how to aggregate a set of predictions into a final classification of the instance. Many of such aggregation strategies have been proposed but a theoretical foundation is missing. Therefore, the fourth research question reads as follows.

> **Research question 4**: Can we develop an aggregation strategy that is feasible in practice and shown to be optimal under reasonable conditions?

To answer our last research question, we presented two important contributions in Chapter 6. First, the strategy of adaptive voting has been derived in the formal setting of label ranking. Under our model assumptions, it is provably optimal in the sense of yielding a MAP prediction of the class label of a test instance. Second, we offered hitherto missing theoretical arguments in favour of the simple weighted voting as a quasi-optimal aggregation strategy in pairwise classification. Since the aggregation strategies can be understood easily, and because of our theoretical and experimental results, we believe that they can be used safely in practice in the sense that other strategies will not deliver much better results (at least, if the results are better anyway).

7.2 Answer to the Problem Statement

In this section, we provide an answer to the problem statement. Our answer is based on the answers to the four research questions as presented in the previous section.

> **Problem statement**: To what extent can machine learning classifiers be used to increase the effectiveness and efficiency of law enforcement?

Our problem statement was motivated by the observation that applying machine learning in law enforcement is far from trivial on a technical and ethical level. We formulated three domain-specific problems in Chapter 1 for which we believe that they strongly hinder the use of machine learning classifiers, namely: high error costs, privacy violations, and evolving class and cost distributions. We associated these problems with four interesting applications of classifiers. Some of these applications already exist in practice, others are difficult to implement due to the severity of the problems to which they are associated. More specifically, we have mentioned the following four applications, with increasing difficulty to be implemented: offender residence prediction, fraud detection, profiling, and predicting recidivism risk.

An affirmative answer to the problem statement can be given when we are able to show that the research presented in the thesis results in approaches that alleviate or prevent all or some of the three domain-specific problems. This allows for a safe implementation of the aforementioned applications, and of course, in accordance with them the implementation of many other applications will follow.

From the affirmative answers to RQ 1 and RQ 2, we may state that we have presented a general approach to learn AUC-optimising decision trees. Next to the benefits of using decision trees in contrast to black-box models, we clearly established that the ranking setting reduces various costs such as waste of financial and human resources, as well as other costs associated with incorrect predictions (including privacy violations). Therefore, we may conclude that the research so far already results in a safer use of machine learning since two out of the three problems are successfully addressed. Offender residence prediction and fraud detection can be efficiently applied due to these results.

In addition, from the answer to RQ 3, we have obtained an approach to construct reliable classifiers. Such classifiers overcome all three domain-specific problems and can be seen as a first important step toward classifiers that output legally correct predictions (a domain expert defines what is legally acceptable or correct by setting various parameters of the approach). This result considerably alleviates the problems associated with profiling and predicting recidivism risk. When implemented in a correct juridical framework, we believe that a safe implementation of these applications is possible.

Finally, our research dealing with multi-class classification problems (RQ 4) resulted in an aggregation strategy that is provably optimal. So clearly, also the answer to RQ 4 will facilitate the use of machine learning classifiers in law enforcement. It opens the door to many applications that deal with more than two class labels. Based on all these answers to the research questions, we may formulate the following conclusive answer to the problem statement.

> **Conclusive answer**: The research presented in this thesis has resulted in techniques that guarantee a safer and more reliable use of machine learning classifiers in law enforcement than is possible so far. Various formal and empirical analyses revealed approaches that will result in a large step forwards toward the implementation of intelligence led policing, eventually resulting in an increased effectiveness and efficiency of law enforcement.

We remark that our positive conclusive answer has a technological nature and only provides indications and suggestions for the world of lawyers and legislators. It is clear that there may still remain (largely normative) difficulties when machine learning is applied to some very critical applications. In the next section, we present four areas of future research to continue improving the results presented in the thesis.

7.3 Future Work

The research presented in this thesis indicates several important and promising areas of future work. In this section, we mention four of the most interesting areas. Our explanation is divided by thesis chapter that answered our research questions.

- *Optimising an approximation of the AUC.* Our analysis about the AUC and its variants has important implications to research dealing with learning AUC-

optimising classifiers. We mentioned two general directions for such classifiers in Chapter 3, namely quadratic programming problems (i.e., AUC-SVMs) and gradient descent approaches. The former was outperformed by the latter. However, a disadvantage of the gradient descent approaches is that they are restricted to learning hyperplanes in instance space, or instances have to be mapped to a space of higher dimensionality. Also, when the objective function resembles the AUC too closely, the gradient might become numerically unstable. Therefore, it would be interesting to extend AUC-SVMs to incorporate proper modifier functions such as the one used by softAUC. Admittedly, improvements are likely to be small, but may be consistent.

- *Fuzzy decision trees for ranking.* The perturbation method for decision trees was proposed in Chapter 4 to substantiate our conjectures. Nonetheless, the method by itself seems to be quite interesting. It is the *dual* of the perturb-and-combine methods, in which the test instance is kept fixed but propagated through more than one decision tree in order to aggregate over a set of predictions (Breiman, 1998). There exists a closed-form solution of the dual method that approaches the final score when the number of propagations through the tree goes to infinity, under the assumption that each feature is only tested once on a path in the tree (Geurts and Wehenkel, 2005). However, instead of propagating a test instance in one of the successors of an internal node, the instance could be directly propagated to all successors with different probabilities. This is ideal to represent model uncertainty around the split value of the feature. Not surprisingly, there is interest in applying fuzzy membership functions for this purpose, resulting in a fuzzy decision tree; see for example Olaru and Wehenkel (2003) and references therein. Experimental results so far only report on accuracy values, thus it would be of interest to explore fuzzy decision tree learning methods in terms of AUC. We conjecture that, when good combination methods for the membership function values on a path are used, fuzzy trees are good rankers. The first results supporting our conjecture are recently given by Hüllermeier and Vanderlooy (2009b).

- *Extra validations for the ROC isometrics approach.* It was shown formally and empirically in Chapter 5 that ROC isometrics can be used to construct reliable classifiers. We considered four performance metrics that are often used in practice, and we had to do an analysis for each of these metrics separately. Therefore, an important area of future research is to investigate whether we can find characteristics of metrics so that our findings can be generalised to other performance metrics. We believe that a geometrical analysis of the isometrics themselves may prove useful for this purpose. In addition, as in related work, we assumed that the empirical ROC curve is a sufficient good approximation of the true curve. Confidence bands for the empirical curve provide a good indication whether this assumption is valid for the data set at hand, but strong theoretical results are desired. The most natural form for such results are upper or lower bounds on the expected performance metric, taking into account its empirical value and the "violation level" of the assumption. Finally, to

construct reliable classifiers, it would be of interest to calibrate the output of classifiers such that the score is a good approximation of the true conditional class probability. Then, the Bayes framework and its extensions can be used. Several calibration methods have been proposed, but a thorough formal and empirical analysis and comparison is lacking.

- *Extending adaptive voting.* The adaptive voting strategy presented in Chapter 6 can be improved by relaxing some model assumptions mainly needed to adhere to corresponding properties of weighted voting. For example, one could think of incorporating prior probabilities in the estimation of the conditional class probabilities p_{ij}, and using asymmetric distributions to model the scores produced by a classifier $\mathcal{M}_{ij}$ (i.e., the distribution of scores given that $\lambda_i \succ_{\boldsymbol{x}} \lambda_j$ is not necessarily the same, except for reflection, as the distribution of scores given that $\lambda_j \succ_{\boldsymbol{x}} \lambda_i$). Also, the maximum likelihood approach for estimating the strengths of base classifiers is susceptible to over-fitting, so using other estimation techniques might be advisable. Besides these directions for improving adaptive voting, we note that the setting of label ranking that we have used for our analysis is interesting in its own right and may provide the basis for additional developments. For example, it allows for studying problems outside the scope of the classification learning setting. We mention the problem of multi-label classification and of course label ranking itself. Finally, we note that, since adaptive voting outputs conditional class probabilities, it can be considered as a calibration method. A comparison with other calibration methods would therefore be of interest.

References

Agarwal, Shivani, Graepel, Thore, Herbrich, Ralf, Har-Peled, Sariel, and Roth, Dan (2005). Generalization Error Bounds for the Area Under the ROC curve. *Journal of Machine Learning Research*, Vol. 6, No. Apr, pp. 393–425.

Ailon, Nir and Mohri, Mehryar (2008). An Efficient Reduction of Ranking to Classification. *Machine Learning*, Vol. 29, No. 2, pp. 103–130.

Airports Council International (2008). *World Airport Traffic Report*. Documents available for free download.

Alpaydin, Ethem (2004). *Introduction to Machine Learning*. MIT Press, New York, NY, USA.

Alvarez, Isabelle, Bernard, Stephan, and Deffuant, Guillaume (2007). Keep the Decision Tree and Estimate the Class Probabilities Using its Decision Boundary. *Proceedings of the 20th International Joint Conference on Artificial Intelligence* (ed. Manuela Veloso), pp. 654 – 659, Springer, Hyderabad, India.

Andrews, Don, Bonta, James, and Wormith, Stephen (2006). The Recent Past and Near Future of Risk and/or Need Assessment. *Crime and Delinquency*, Vol. 52, No. 1, pp. 7–27.

Asuncion, Arthur and Newman, David (2007). *UCI Machine Learning Repository*. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Atiya, Amir (2005). Estimating the Posterior Probabilities Using the K-Nearest Neighbor Rule. *Neural Computation*, Vol. 17, No. 3, pp. 731–740.

Auerhahn, Kathleen (1999). Selective Incapacitation and the Problem of Prediction. *Criminology*, Vol. 37, No. 4, pp. 703–734.

Bibel, Wolfgang (2004). AI and the Conquest of Complexity of Law. *Artificial Intelligence and Law*, Vol. 12, No. 3, pp. 159–180.

Bickel, Peter, Ritov, Yaacov, and Zakai, Alon (2006). Some Theory for Generalized Boosting Algorithms. *Journal of Machine Learning Research*, Vol. 2, No. May, pp. 705–732.

Bishop, Christopher (2007). *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA.

Blokland, Arjan and Nieuwbeerta, Paul (2005). The Effects of Life Circumstances on Longitudinal Trajectories of Offending. *Criminology*, Vol. 43, No. 4, pp. 1203–1240.

Blokland, Arjan and Nieuwbeerta, Paul (2006). De Consequenties van "Three Strikes You're Out" in Nederland: Kosten en Baten van Het Selectief Detineren van Veelplegers. *Proces*, Vol. 85, No. 4, pp. 124–130. (in Dutch).

Blokland, Arjan (2005a). *Crime over the Life Span: Trajectories of Criminal Behavior in Dutch Offenders*. Ph.D. thesis, Universiteit Leiden, Leiden, The Netherlands.

Blokland, Arjan, Nagin, Daniel, and Nieuwbeerta, Paul (2005b). Life Span Offending Trajectories of a Dutch Conviction Cohort. *Criminology*, Vol. 43, No. 4, pp. 919–954.

Bolton, Richard and Hand, David (2002). Statistical Fraud Detection: A Review. *Statistical Science*, Vol. 17, No. 3, pp. 233–255.

Bradley, Andrew (1997). The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern recognition*, Vol. 30, No. 7, pp. 1145–1159.

Brefeld, Ulf and Scheffer, Tobias (2005). AUC Maximizing Support Vector Learning. *Proceedings of the 2nd Workshop on ROC Analysis in Machine Learning* (eds. Cèsar Ferri, Nicolas Lachiche, Sofus Macskassy, and Alain Rakotomamonjy), ACM, Bonn, Germany.

Breiman, Leo, Friedman, Jerome, Stone, Charles, and Olshen, Richard (1984). *Classification and Regression Trees*. Chapmand and Hall/CRC Press, Boca Raton, FL, USA.

Breiman, Leo (1998). Arcing Classifiers. *The Annals of Statistics*, Vol. 26, No. 3, pp. 801–849.

Calders, Toon and Jaroszewicz, Szymon (2007). Efficient AUC Optimization for Classification. *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases* (eds. Joost Kok, Jacek Koronacki, Ramon López de Mántaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron), pp. 42–53, Springer, Warsaw, Poland.

Carter, David and Lansing, East (2007). Civil Rights and Law Enforcement Intelligence. *The Police Chief*, Vol. 74, No. 6. (page numbers not available).

Caruana, Rich and Niculescu-Mizil, Alexandru (2004). An Empirical Evaluation of Supervised Learning for ROC Area. *Proceedings of the 1st International Workshop on ROC Analysis in Artificial Intelligence* (eds. José Hernández-Orallo, Cèsar Ferri, Nicolas Lachiche, and Peter Flach), pp. 1–8, IOS Press, Valencia, Spain.

Caruana, Rich and Niculescu-Mizil, Alexandru (2006). An Empirical Comparison of Supervised Learning Algorithms. *Proceedings of the 23rd International Conference on Machine Learning* (eds. William Cohen and Andrew Moore), pp. 161–168, ACM, Pittsburgh, PA, USA.

Cestnik, Bojan (1990). Estimating Probabilities: A Crucial Task in Machine Learning. *Proceedings of the 9th European Conference on Artificial Intelligence* (ed. Luigia Aiello), pp. 147–149, Pitman Publishing, Stockholm, Sweden.

Chakrabarti, Samidh and Strauss, Aaron (2002). Carnival Booth: An Algorithm for Defeating the Computer-Assisted Passenger Screening System. *First Monday*, Vol. 7, No. 10. (page numbers not available).

Chow, Chi (1957). An Optimum Character Recognition System using Decision Functions. *IRE Transactions on Electronic Computers*, Vol. 6, No. 4, pp. 247–254.

Chow, Chi (1970). On Optimum Recognition Error and Reject Tradeoff. *IEEE Transactions on Information Theory*, Vol. 16, No. 1, pp. 41–46.

Clémençon, Stéfan and Vayatis, Nicolas (2008). Approximation of the Optimal ROC Curve and a Tree-based Ranking Algorithm. *Proceedings of the 19th International Conference on Algorithmic Learning Theory* (eds. Yoav Freund, László Györfi, György Turán, and Thomas Zeugmann), pp. 22–37, Springer, Budapest, Hungary.

Cohen, Ira and Goldszmidt, Moisés (2004). Properties and Benefits of Calibrated Classifiers. *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases* (eds. Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi), pp. 125–136, Springer, Pisa, Italy.

Cohen, William, Shapire, Robert, and Singer, Yoram (1997). Learning to Order Things. *Advances in Neural Information Processing Systems 10* (eds. Michael Jordan, Micheal Kearns, and Sara Solla), pp. 451–457, MIT Press, Denver, CO, USA.

Cope, Nina (2004). Intelligence Led Policing or Policing Led Intelligence? Integrating Volume Crime Analysis into Policing. *British Journal of Criminology*, Vol. 44, No. 2, pp. 188–203.

Corani, Giorgio and Zaffalon, Marco (2008). Learning Reliable Classifiers From Small or Incomplete Data Sets: the Naive Credal Classifier 2. *Journal of Machine Learning Research*, Vol. 9, No. Apr, pp. 581–621.

Cortes, Corinna and Mohri, Mehryar (2003). AUC Optimization vs. Error Rate Minimization. *Advances in Neural Information Processing Systems 16* (eds. Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf), MIT Press, Vancouver, BC, Canada.

Cover, Thomas and Hart, Peter (1967). Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, Vol. 13, No. 1, pp. 21–27.

Crammer, Koby and Singer, Yoram (2003). Ultraconservative Online Algorithms for Multiclass Problems. *Journal of Machine Learning Research*, Vol. 2, No. Jan, pp. 951–991.

Cutzu, Florin (2003). Polychotomous Classification with Pairwise Classifiers: A New Voting Principle. *Proceedings of the 4th International Workshop on Multiple Classifier Systems* (eds. Terry Windeatt and Fabio Roli), pp. 115–124, Springer, Guilford, UK.

De Hert, Paul, Huisman, Wim, and Vis, Thijs (2005). Intelligence Led Policing Ontleed. *Tijdschrift voor Criminologie*, Vol. 47, No. 4, pp. 365–376. (in Dutch).

Demšar, Janes (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, Vol. 7, No. Jan, pp. 1–30.

Devlin, Keith and Lorden, Gary (2007). *The Numbers Behind Numb3rs: Solving Crime with Mathematics*. Penguin Group, New York, NY, USA.

Dietterich, Thomas and Bakiri, Ghulum (1995). Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research*, Vol. 2, No. 1, pp. 263–286.

Domingos, Pedro and Pazzani, Michael (1997). On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning*, Vol. 29, No. 2, pp. 103–130.

Elkan, Charles (2001). The Foundations of Cost-Sensitive Learning. *Proceedings of the 17th International Joint Conference on Artificial Intelligence* (ed. Bernhard Nebel), pp. 973–978, Morgan Kaufmann, Seattle, WA, USA.

Esposito, Floriana, Malerba, Donato, and Semeraro, Giovanni (1997). A Comparative Analysis of Methods for Pruning Decision Trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 5, pp. 476–491.

Fawcett, Tom and Flach, Peter (2005). A Response to Webb and Ting's On the Application of ROC Analysis to Predict Classification Performance Under Varying Class Distributions. *Machine Learning*, Vol. 58, No. 1, pp. 33–38.

Fawcett, Tom and Niculescu-Mizil, Alexandru (2007). PAV and the ROC Convex Hull. *Machine Learning*, Vol. 68, No. 1, pp. 97–106.

Fawcett, Tom (2003). ROC Graphs: Notes and Practical Considerations for Researchers. Technical Report HPL-2003-4, HP Laboratories.

Fawcett, Tom (2006). ROC Graphs with Instance-Varying Costs. *Pattern Recognition Letters*, Vol. 27, No. 8, pp. 882–891.

Feeley, Malcolm and Simon, Jonathan (2003). The New Penology. *Criminological Perspectives, Essential Readings*, pp. 434–446. Sage Publications, London, UK.

Ferri, Cèsar, Flach, Peter, and Hernández-Orallo, José (2002). Learning Decision Trees Using the Area Under the ROC Curve. *Proceedings of the 19th International Conference on Machine Learning* (eds. Claude Sammut and Achim Hoffmann), pp. 139–146, Morgan Kaufmann, Sydney, Australia.

Ferri, Cèsar, Flach, Peter, and Hernández-Orallo, José (2003). Improving the AUC of Probabilistic Estimation Trees. *Proceedings of the 14th European Conference on Machine Learning* (eds. Nada Lavrac, Dragan Gamberger, Ljupco Todorovski, and Hendrik Blockeel), pp. 121–132, Springer, Cavtat-Dubrovnik, Croatia.

Ferri, Cèsar, Flach, Peter, and Hernández-Orallo, José (2004). Delegating Classifiers. *Proceedings of the 1st International Workshop on ROC Analysis in Artificial Intelligence* (eds. José Hernández-Orallo, Cèsar Ferri, Nicolas Lachiche, and Peter Flach), pp. 37–44, IOS Press, Valencia, Spain.

Ferri, Cèsar, Flach, Peter, Hernández-Orallo, José, and Senad, Athmane (2005). Modifying ROC Curves to Incorporate Predicted Probabilities. *Proceedings of the 2nd Workshop on ROC Analysis in Machine Learning* (eds. Cèsar Ferri, Nicolas Lachiche, Sofus Macskassy, and Alain Rakotomamonjy), ACM, Bonn, Germany.

Flach, Peter and Wu, Shaomin (2005). Repairing Concavities in ROC Curves. *Proceedings of the 19th International Joint Conference on Artificial Intelligence* (eds. Leslie Kaelbling and Alessandro Saffiotti), pp. 702–707, Professional Book Center, Edinburgh, UK.

Flach, Peter (2003). The Geometry of ROC Space: Understanding Machine Learning Metrics through ROC Isometrics. *Proceedings of the 20th International Conference on Machine Learning* (eds. Tom Fawcett and Nina Mishra), pp. 194–201, AAAI Press, Washington, DC, USA.

Fodor, János and Roubens, Marc (1994). *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Frélicot, Carl and Mascarilla, Laurent (2002). Reject Strategies Driven Combination of Pattern Classifiers. *Pattern Analysis and Applications*, Vol. 5, No. 2, pp. 234–243.

Friedel, Caroline, Rückert, Ulrich, and Kramer, Stefan (2006). Cost Curves for Abstaining Classifiers. *Proceedings of the ICML 2006 Workshop on ROC Analysis* (eds. Nicolas Lachiche, Cèsar Ferri, and Sofus Macskassy), pp. 33–40, ACM, Pittsburgh, PA, USA.

Friedman, Jerome (1996). Another Approach to Polychotomous Classification. Technical report, Department of Statistics, Stanford University.

Friedman, Jerome (1997). On Bias, Variance, 0/1-Loss, and the Curse of Dimensionality. *Data Mining and Knowledge Discovery*, Vol. 1, No. 1, pp. 55–77.

Fumera, Giorgio, Roli, Fabio, and Giacinto, Giorgio (2000). Multiple Reject Thresholds for Improving Classification Reliability. *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition* (eds. Francesc Ferri, José Iñesta, Adnan Amin, and Pavel Pudil), pp. 863–871, Springer, Alicante, Spain.

Fürnkranz, Johannes and Flach, Peter (2005). ROC 'n' Rule Learning – Towards a Better Understanding of Covering Algorithms. *Machine Learning*, Vol. 58, No. 1, pp. 39–77.

Fürnkranz, Johannes and Hüllermeier, Eyke (2003). Pairwise Preference Learning and Ranking. *Proceedings of the 13th European Conference on Machine Learning* (eds. Nada Lavrac, Dragan Gamberger, Ljupco Todorovski, and Hendrik Blockeel), pp. 145–156, Springer, Cavtat-Dubrovnik, Croatia.

Fürnkranz, Johannes (2001). Round Robin Rule Learning. *Proceedings of the 18th International Conference on Machine Learning* (eds. Carla Brodley and Andrea Danyluk), pp. 146–153, Morgan Kaufmann, Williamstown, MA, USA.

Fürnkranz, Johannes (2002). Round Robin Classification. *Journal of Machine Learning Research*, Vol. 2, No. Mar, pp. 721–747.

Fürnkranz, Johannes (2003). Round Robin Ensembles. *Intelligent Data Analysis*, Vol. 7, No. 5, pp. 385–404.

Geurts, Pierre and Wehenkel, Louis (2005). Closed-Form Dual Perturb and Combine for Tree-Based Models. *Proceedings of the 22th International Conference on Machine Learning* (eds. Luc De Raedt and Stefan Wrobel), pp. 233–240, ACM, Bonn, Germany.

Geurts, Pierre, Ernst, Damien, and Wehenkel, Louis (2006). Extremely Randomized Trees. *Machine Learning*, Vol. 36, No. 1, pp. 3–42.

Gill, Peter and Phythian, Mark (2006). *Intelligence in an Insecure World.* Polity Press, Cambrigde, UK.

Gill, Peter (2000). *Rounding Up the Usual Suspects?: Developments in Contemporary Law Enforcement.* Ashgate Publishers, Surrey, UK.

Goodwin, Maurice (2007). *Hunting Serial Predators*. Jones and Bartlett Publishers, Toronto, Ontario, Canada, 2nd edition.

Guyon, Isabelle (2007). *Data Representation Discovery Workshop of the 20th International Joint Conference on Neural Networks*. http://clopinet.com/isabelle/Projects/agnostic/.

Ha, Thien (1996a). An Experimental Study of the Optimal Class-Selective Rejection Rule. Technical Report IAM-96-008, Institute of Computer Science and Applied Mathematics, University of Berne, Switzerland.

Ha, Thien (1996b). On Functional Relation between Class-Selective Rejection Error and Average Number of Classes. *IEEE International Joint Symposia on Intelligence and Systems*, Vol. 4-5, pp. 282–282.

Ha, Thien (1997). The Optimum Class-Selective Rejection Rule. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 6, pp. 608–615.

Hand, David and Till, Robert (2001). A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Machine Learning*, Vol. 45, No. 2, pp. 171–186.

Hanley, James and McNeil, Barbara (1982). The Meaning and Use of the Area Under a Receiver Operator Characteristic ROC Curve. *Radiology*, Vol. 143, No. 1, pp. 29–36.

Har-Peled, Sariel, Roth, Dan, and Zimak, Dav (2002). Constraint Classification for Multiclass Classification and Ranking. *Advances in Neural Information Processing Systems 15* (eds. Suzanna Becker, Sebastian Thrun, and Klaus Obermayer), pp. 785–792, MIT Press, Vancouver, BC, Canada.

Hastie, Trevor and Tibshirani, Robert (1998). Classification by Pairwise Coupling. *The Annals of Statistics*, Vol. 26, No. 2, pp. 451–471.

Hastie, Trevor, Tibshirani, Robert, and Friedman, Jerome (2001). *The Elements of Statistical Learning*. Springer, New York, NY, USA.

Herschtal, Alan and Raskutti, Bhavani (2004). Optimising Area under the ROC Curve using Gradient Descent. *Proceedings of the 21st International Conference on Machine Learning* (ed. Carla Brodley), ACM, Banff, AB, Canada.

Hsu, Chih-Wei and Lin, Chih-Jen (2002). A Comparison of Methods for Multiclass Support Vector Machines. *IEEE Transactions on Neural Networks*, Vol. 13, No. 2, pp. 415–425.

Huang, Jin and Ling, Charles (2005). Using AUC and Accuracy in Evaluating Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 3, pp. 299–310.

Huang, Jin and Ling, Charles (2006). Evaluating Model Selection Abilities of Performance Measures. *Proceedings of the 1st Workshop on Evaluation Methods for Machine Learning* (eds. Chris Drummond, William Elazmeh, and Nathalie Japkowicz), pp. 12–17, AAAI Press, Boston, MA, USA.

Huang, Jin, Ling, Charles, Zhang, Harry, and Matwin, Stan (2008). Proper Model Selection with Significance Test. *Proceedings of the 19th European Conference on Machine Learning* (eds. Walter Daelemans, Bart Goethals, and Katharina Morik), pp. 536–547, Springer, Antwerp, Belgium.

Hühn, Jens and Hüllermeier, Eyke (2009). FR3: A Fuzzy Rule Learner for Inducing Reliable Classifiers. *IEEE Transactions on Fuzzy Systems*, Vol. 17, No. 1, pp. 138–149.

Hüllermeier, Eyke and Brinker, Klaus (2008). Learning Valued Preference Structures for Solving Classification Problems. *Fuzzy Sets and Systems*, Vol. 159, No. 18, pp. 2337–2352.

Hüllermeier, Eyke and Fürnkranz, Johannes (2004). Comparison of Ranking Procedures in Pairwise Preference Learning. *Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knoweldge-Based Systems* (eds. Bernadette Bouchon-Meunier, Giulianella Coletti, and Ronald Yager), pp. 535–542, Springer, Perugua, Italy.

Hüllermeier, Eyke and Fürnkranz, Johannes (2007). On Minimizing the Position Error in Label Ranking. *Proceedings of the 18th European Conference on Machine Learning* (eds. Joost Kok, Jacek Koronacki, Ramon López de Mántaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron), pp. 583–590, Springer, Warsaw, Poland.

Hüllermeier, Eyke and Fürnkranz, Johannes (2009). *On Predictive Accuracy and Risk Minimization in Pairwise Label Ranking.* Journal of Computer and System Sciences (accepted).

Hüllermeier, Eyke and Vanderlooy, Stijn (2008a). An Empirical and Formal Analysis of Decision Trees for Ranking. Technical Report Computer Science Series 56, Philipps Universität Marburg, Germany.

Hüllermeier, Eyke and Vanderlooy, Stijn (2008b). Weighted Voting as Approximate MAP Prediction in Pairwise Classification. *Proceedings of the LWA Workshop on Knowledge Discovery, Data Mining, and Machine Learning (KDML 2008)* (eds. Joachim Baumeister and Martin Atzmüller), pp. 34–41, University of Würzburg, Würzburg, Germany.

Hüllermeier, Eyke and Vanderlooy, Stijn (2009a). *Combining Predictions in Pairwise Classification: An Optimal Adaptive Voting Strategy and its Relation to Weighted Voting.* Pattern Recognition (under revision).

Hüllermeier, Eyke and Vanderlooy, Stijn (2009b). *Why Fuzzy Trees are Good Rankers*. IEEE Transactions on Fuzzy Systems (accepted).

Hüllermeier, Eyke (2004). Instance-Based Prediction with Guaranteed Confidence. *Proceedings of the 16th European Conference on Artificial Intelligence* (eds. Ramon López de Mántaras and Lorenza Saitta), pp. 97–101, IOS Press, Valencia, Spain.

Hüllermeier, Eyke, Fürnkranz, Johannes, Cheng, Weiwei, and Brinker, Klaus (2008). Label Ranking by Learning Pairwise Preferences. *Artificial Intelligence*, Vol. 172, Nos. 16–17, pp. 1897–1916.

International Crime Victims Survey (2008). *Criminal Victimisation in International Perspective*. Documents available for free download.

Jans, Mieke, Lybaert, Nadine, and Vanhoof, Koen (2006). Data Mining for Fraud Detection: Toward an Improvement on Internal Control Systems? *Proceedings of the 7th International Research Symposium on Accounting Information Systems* (ed. Steve Sutton), pp. 41–58, Association for Information Systems.

Jiang, Liangxiao, Zhang, Harry, and Su, Jiang (2005). Learning k-Nearest Neighbor Naive Bayes for Ranking. *Proceedings of the 1st International Conference on Advanced Data Mining and Applications* (eds. Xue Li, Shuliang Wang, and Zhao Dong), pp. 175–185, Springer, Wuhan, China.

Jonas, Jeff and Harper, Jim (2006). Effective Counterterrorism and the Limited Role of Predictive Data Mining. *Police Analysis*, Vol. 584, No. Dec, pp. 1–12.

Kielman, Hugo and Koelewijn, Wouter (2005a). Meer Maatregelen, Minder Risico. *IT Monitor*, Vol. 8, No. Aug, pp. 10–11. (in Dutch).

Kielman, Hugo and Koelewijn, Wouter (2005b). Minder Registers, Meer Gegevens. *Ars Aequi (Mens en Maatschappij)*, Vol. 6, No. Jun, pp. 451–457. (in Dutch).

Kohavi, Ron (1996). Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (eds. Evangelos Simoudis, Jiawei Han, and Usama Fayyad), pp. 202–207, AAAI Press, Portland, OR, USA.

Kurlander, Neil (2005). Fighting Crime and Terrorism through Data Integration. *The Police Chief*, Vol. 72, No. 2. (page numbers not available).

Lavrac, Nada and Dzeroski, Saso (1994). *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, New York, NY, USA.

Lenhardt, Alfonso (2006). The Economics of Prevention: Reducing Costs and Crime. *The Police Chief*, Vol. 73, No. 7. (page numbers not available).

Ling, Charles and Sheng, Victor (2010). Cost-Sensitive Learning and the Class Imbalance Problem. *Encyclopedia of Machine Learning* (ed. Claude Sammut). Springer. (to appear).

Ling, Charles, Huang, Jin, and Zhang, Harry (2003). AUC: A Statistically Consistent and more Discriminating Measure than Accuracy. *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (eds. Georg Gottlob and Toby Walsh), pp. 519–526, AAAI Press.

Lu, Jingli, Yang, Ying, and Webb, Geoffrey (2006). Incremental Discretization for Naive Bayes Classifier. *Proceedings of the 2nd International Conference on Advanced Data Mining and Applications* (eds. Xue Li, Osmar Zaïane, and Zhanhuai Li), pp. 223–238, Springer, Xi'an, China.

Macskassy, Sofus and Provost, Foster (2004). Confidence Bands for ROC Curves: Methods and an Empirical Study. *Proceedings of the 1st International Workshop on ROC Analysis in Artificial Intelligence* (eds. José Hernández-Orallo, Cèsar Ferri, Nicolas Lachiche, and Peter Flach), pp. 61–70, IOS Press, Valencia, Spain.

Mann, Henry and Whitney, Donald (1947). On a Test Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, Vol. 18, No. 1, pp. 50–60.

Marden, John (1995). *Analyzing and Modeling Rank Data*. Chapman & Hall, London, UK.

Marrocco, Claudio, Molinara, Mario, and Tortorella, Francesco (2007). An Empirical Comparison of Ideal and Empirical ROC-Based Reject Rules. *Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition* (ed. Petra Perner), pp. 47–60, Springer, Leipzig, Germany.

Mason, S. and Graham, N. (2002). Areas Beneath the ROC and ROL Curves: Statistical Significance and Interpretation. *Quarterly Journal of the Royal Meteorological Society*, Vol. 128, No. 584, pp. 2145–2166.

McCue, Colleen (2003). Connecting the Dots: Data Mining and Predictive Analytics in Law Enforcement and Intelligence Analysis. *The Police Chief*, Vol. 70, No. 10. (page numbers not available).

McNeil, Barbara and Hanley, James (1984). Statistical Approaches to the Analysis of Receiver Operating Characteristic (ROC) Curves. *Medical Decision Making*, Vol. 4, No. 2, pp. 137–150.

Mitchell, Tom (1997). *Machine Learning*. McGraw-Hill, New York, NY, USA.

Moffitt, Terrie (2006). Life-Course-Persistent and Adolescence-Limited Anti-Social Behavior. *Developmental Psychopathology* (eds. Dante Cicchetti and Donald Cohen), Vol. 3, pp. 570–598. Wiley, New York, NY, USA, 2nd edition.

Moreira, Miguel and Mayoraz, Eddy (1998). Improved Pairwise Coupling Classification with Correcting Classifiers. *Proceedings of the 10th European Conference on Machine Learning* (eds. Claire Nedellec and Céline Rouveirol), pp. 160–171, Springer, Chemnitz, Germany.

Muzzolini, Russell, Yang, Yee-Hong, and Pierson, Roger (1998). Classifier Design with Incomplete Knowledge. *Pattern Recognition*, Vol. 31, No. 4, pp. 345–369.

Niculescu-Mizil, Alexandru and Caruana, Rich (2005). Predicting Good Probabilities with Supervised Learning. *Proceedings of the 22nd International Conference on Machine Learning* (eds. Luc De Raedt and Stefan Wrobel), pp. 625–632, ACM Press, Bonn, Germany.

Obuchowski, Nancy, Sergey, Berbaum, Beiden Kevin, Hillis, Stephen, Ishwaran, Hemant, Song, Hae, and Wagner, Robert (2004). Multireader, Multicase Receiver Operating Characteristic Analysis: an Empirical Comparison of Five Methods. *Academic Radiology*, Vol. 11, No. 9, pp. 980–995.

Olaru, Cristina and Wehenkel, Louis (2003). A Complete Fuzzy Decision Tree Technique. *Fuzzy Sets and Systems*, Vol. 138, No. 2, pp. 221–254.

Oskamp, Anja and Lauritsen, Marc (2002). AI in Law Practice? So Far, Not Much. *Artificial Intelligence and Law*, Vol. 10, No. 4, pp. 227–236.

Pietraszek, Tadeusz (2007a). Classification of Intrusion Detection Alerts using Abstaining Classifiers. *Intelligent Data Analysis*, Vol. 11, No. 3, pp. 293–316.

Pietraszek, Tadeusz (2007b). On the Use of ROC Analysis for the Optimization of Abstaining Classifiers. *Machine Learning*, Vol. 68, No. 2, pp. 137–169.

Provost, Foster and Domingos, Pedro (2003). Tree-induction for Probability Based Ranking. *Machine Learning*, Vol. 52, No. 3, pp. 199–215.

Provost, Foster and Fawcett, Tom (2001). Robust Classification for Imprecise Environments. *Machine Learning*, Vol. 42, No. 3, pp. 203–231.

Provost, Foster and Kolluri, Venkateswarlu (1999). A Survey of Methods for Scaling Up Inductive Algorithms. *Data Mining and Knowledge Discovery*, Vol. 3, No. 2, pp. 131–169.

Provost, Foster, Fawcett, Tom, and Kohavi, Ron (1998). The Case Against Accuracy Estimation for Comparing Induction Algorithms. *Proceedings of the 15th International Conference on Machine Learning* (ed. Jude Shavlik), pp. 43–48, Morgan Kaufmann, Madison, WI, USA.

Quinlan, Ross (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, CA, USA.

Quost, Benjamin, Denoeux, Thierry, and Masson, Marie-Hélène (2007). Pairwise Classifier Combination using Belief Functions. *Pattern Recognition Letters*, Vol. 28, No. 5, pp. 644–653.

Rakotomamonjy, Alain (2004). Optimizing Area Under ROC Curve with SVMs. *Proceedings of the 1st Workshop on ROC Analysis and Artificial Intelligence* (eds. José Hernández-Orallo, Cèsar Ferri, Nicolas Lachiche, and Peter Flach), pp. 71–80, IOS Press, Valencia, Spain.

Ratcliffe, Jerry (2008). *Intelligence Led Policing*. Willan Publishing, Portland, OR, USA.

Rosset, Saharon (2004). Model Selection via the AUC. *Proceedings of the 21st International Conference on Machine Learning* (ed. Carla Brodley), pp. 89–97, ACM, Banff, AB, Canada.

Rossmo, Kim (1999). *Geographic Profiling*. CRC Press, Florida, FL, USA.

Santos-Pereira, Carla and Pires, Ana (2005). On Optimal Reject Rules and ROC Curves. *Pattern Recognition Letters*, Vol. 26, No. 7, pp. 943–952.

Schwalbe, Craig, Fraser, Mark, Day, Steven, and Cooley, Valerie (2006). Classifying Juvenile Offenders According to Risk of Recidivism. *Criminal Justice and Behaviour*, Vol. 33, No. 3, pp. 305–324.

Skalak, David, Niculescu-Mizil, Alexandru, and Caruana, Rich (2007). Classifier Loss Under Metric Uncertainty. *Proceedings of the 18th European Conference on Machine Learning* (eds. Joost Kok, Jacek Koronacki, Ramon López de Mántaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron), pp. 310–322, Springer, Warsaw, Poland.

Smyth, Padhraic, Gray, Alexander, and Fayyad, Usama (1995). Retrofitting Decision Tree Classifiers Using Kernel Density Estimation. *Proceedings of the 12th International Conference on Machine Learning* (eds. Armand Prieditis and Stuart Russell), pp. 506–514, Morgan Kaufmann, Tahoe City, CA, USA.

Steck, Harald (2007). Hinge Rank Loss and the Area Under the ROC Curve. *Proceedings of the 18th European Conference on Machine Learning* (eds. Joost Kok, Jacek Koronacki, Ramon López de Mántaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron), pp. 347–358, Springer, Warsaw, Poland.

Swets, John (1964). *Signal Detection and Recognition by Human Observers*. Peninsula Publising, Newport Beach, CA, USA.

Swets, John, Dawes, Robyn, and Monahan, John (2000). Better Decisions through Science. *Scientific American*, Vol. 283, No. 4, pp. 82–87.

Tax, David and Veenman, Cor (2005). Tuning the Hyperparameter of an AUC-Optimized Classifier. *Proceedings of the 17th Belgium-Netherlands Conference on Artificial Intelligence* (eds. Katja Verbeeck, Karl Tuyls, Ann Nowe, Bernard Manderick, and Bart Kuijpers), pp. 224–231, Royal Flemish Academy of Belgium for Science and Arts, Brussels, Belgium.

Tax, David, Duin, Robert, and Arzhaeva, Yulia (2006). Linear Model Combining by Optimizing the Area under the ROC Curve. *Proceedings of the 18th International Conference on Pattern Recognition* (eds. Yuan Tang, Patrick Wang, Guy Lorette, Daniel Yeung, and Hong Yan), pp. 119–122, IEEE Computer Society Press, Hong Kong, China.

Thibault, Edward, Lynch, Lawrence, and McBride, Bruce (2006). *Proactive Police Management*. Prentice Hall, London, UK, 7th edition.

Tilley, Nick (2005). Community Policing, Problem-Oriented Policing and Intelligence-Led Policing. *Handbook of Policing*, pp. 311–339. William Publishing, Pleymouth Devon.

Ting, Kai (2002). A Study of the Effect of Class Distribution using Cost-Sensitive Learning. *Proceedings of the 5th International Conference on Discovery Science* (eds. Steffen Lange, Ken Satoh, and Carl Smith), pp. 98–112, Springer, Lübeck, Germany.

Tortorella, Francesco (2004). Reducing the Classification Cost of Support Vector Classifiers through an ROC-based Reject Rule. *Pattern Analysis and Applications*, Vol. 7, No. 2, pp. 128–143.

Tortorella, Francesco (2005). A ROC-based Reject Rule for Dichotomizers. *Pattern Recognition Letters*, Vol. 26, No. 2, pp. 167–180.

Transportation Security Administration (2008). *Secure Flight Program*. Documents available for free download.

Van Rijsbergen, Cornelis (1979). *Information Retrieval*. Department of Computer Science, University of Glasgow, 2nd edition.

Vanderlooy, Stijn and Hüllermeier, Eyke (2008). A Critical Analysis of Variants of the AUC. *Machine Learning*, Vol. 72, No. 3, pp. 247–262.

Vanderlooy, Stijn and Sprinkhuizen-Kuyper, Ida (2007). A Comparison of Two Approaches to Classify with Guaranteed Performance. *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases* (eds. Joost Kok, Jacek Koronacki, Ramon López de Mántaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron), LNAI 4702, pp. 288–299, Springer, Warsaw, Poland.

Vanderlooy, Stijn, Verbeek, Joop, and van den Herik, Jaap (2005). Kunstmatige Intelligentie en de Wet Politiegegevens. *Veiligheid en Recht: Nieuwe Doelwitten, Nieuwe Strategieën* (eds. Wim Huisman, Martin Moerings, and Guido Suurmond), pp. 255–266, Boom Juridische Uitgevers, Leiden, the Netherlands. (in Dutch).

Vanderlooy, Stijn, Postma, Eric, Tuyls, Karl, and Sprinkhuizen-Kuyper, Ida (2006a). Reliable Instance Classifications in Law Enforcement. *Proceedings of the 18th Benelux Conference on Artifical Intelligence* (eds. Pierre-Yves Schobbens, Wim Vanhoof, and Gabriel Schwanen), pp. 323–330, Namur, Belgium.

Vanderlooy, Stijn, Sprinkhuizen-Kuyper, Ida, and Smirnov, Evgueni (2006b). An Analysis of Reliable Classifiers through ROC Isometrics. *Proceedings of the ICML 2006 Workshop on ROC Analysis* (eds. Nicolas Lachiche, Cèsar Ferri, and Sofus Macskassy), pp. 55–62, ACM, Pittsburgh, USA.

Vanderlooy, Stijn, Sprinkhuizen-Kuyper, Ida, and Smirnov, Evgueni (2006c). Reliable Classifiers in ROC Space. *Proceedings of the 15th Annual Machine Learning Conference of Belgium and the Netherlands* (eds. Yvan Saeys, Elena Tsiporkova, Bernard De Baets, and Yves Van de Peer), pp. 113–120, Universiteit Gent, Ghent, Belgium.

Vanderlooy, Stijn, Verbeek, Joop, and van den Herik, Jaap (2007). Towards Privacy-Preserving Data Mining in Law Enforcement. *Journal of International Commercial Law and Technology*, Vol. 2, No. 4, pp. 202–210.

Vanderlooy, Stijn, Sprinkhuizen-Kuyper, Ida, Smirnov, Evgueni, and van den Herik, Jaap (2009). The ROC Isometrics Approach to Construct Reliable Classifiers. *Intelligent Data Analysis*, Vol. 13, No. 1, pp. 3–37.

Vilalta, Ricardo and Oblinger, Daniel (2000). A Quantification of Distance Bias Between Evaluation Metrics In Classification. *Proceedings of the 17th International Conference on Machine Learning* (ed. Pat Langley), pp. 1087–1094, Morgan Kaufmann, Stanford, CA, USA.

Vlachos, Pantelis (2008). *StatLib Project Repository*. http://lib.stat.cmu.edu/.

Vovk, Vladimir, Gammerman, Alex, and Shafer, Glenn (2005). *Algorithmic Learning in a Random World*. Springer, New York, NY, USA.

Wang, Bin and Zhang, Harry (2006). Improving the Ranking Performance of Decision Trees. *Proceedings of the 17th European Conference on Machine Learning* (eds. Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou), pp. 461–472, Springer, Berlin, Germany.

Wechsler, Harry (2006). *Reliable Face Recognition Methods*. Springer, New York, NY, USA.

Weiss, Gary and Provost, Foster (2003). Learning when Training Data are Costly: The Effect of Class Distribution on Tree Induction. *Journal of Artificial Intelligence Research*, Vol. 19, pp. 315–354.

Westphal, Christopher (2008). *Data Mining for Intelligence, Fraud & Criminal Detection: Advanced Analytics & Information Sharing Technologies*. CRC Press, Florida, FL, USA.

Witten, Ian and Frank, Eibe (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, CA, USA, 2nd edition.

Wu, Ting-Fan, Lin, Chih-Jen, and Weng, Ruby (2004). Round Robin Classification. *Journal of Machine Learning Research*, Vol. 5, No. 1, pp. 975–1005.

Wu, Shaomin, Flach, Peter, and Ferri, Cèsar (2007). An Improved Model Selection Heuristic for AUC. *Proceedings of the 18th European Conference on Machine Learning* (eds. Joost Kok, Jacek Koronacki, Ramon López de Mántaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron), pp. 478–489, Springer, Warsaw, Poland.

Xu, Lihao, Krzyzak, Adam, and Suen, Chen (1992). Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 22, No. 3, pp. 418–435.

Yan, Lian, Dodier, Robert, Mozer, Michael, and Wolniewicz, Richard (2003). Optimizing Classifier Performance via an Approximation to the Wilcoxon-Mann-Whitney Statistic. *Proceedings of the 20th International Conference on Machine Learning* (eds. Tom Fawcett and Nina Mishra), pp. 848–855, AAAI Press, Washington, DC, USA.

Yen, John and Popp, Robert (2005). AI Technologies for Homeland Security. Technical Report SS-05-01, AAAI Spring Symposium, Stanford, CA, USA.

Zadrozny, Bianca and Elkan, Charles (2001). Obtaining Calibrated Probability Estimates from Decision Trees and Naive Bayesian Classifiers. *Proceedings of the 18th International Conference on Machine Learning* (eds. Carla Brodley and Andrea Pohoreckyj Danyluk), pp. 609–616, Morgan Kaufmann, San Francisco, CA, USA.

Zadrozny, Bianca and Elkan, Charles (2002). Transforming Classifier Scores into Accurate Multiclass Probability Estimates. *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (eds. David Hand, Daniel Keim, and Raymond Ng), pp. 694–699, ACM, Edmonton, AB, Canada.

Zhang, Harry and Su, Jiang (2006). Learning Probabilistic Decision Trees for AUC. *Pattern Recognition Letters*, Vol. 27, No. 8, pp. 892–899.

Zhang, Jian and Yang, Yiming (2004). Probabilistic Score Estimation with Piecewise Logistic Regression. *Proceedings of the 21st International Conference on Machine Learning* (ed. Carla Brodley), pp. 115–123, ACM, Banff, AB, Canada.

Appendix A

Proofs of Theorems about the ROC Isometrics Approach

This appendix is subdivided into eleven sections and each of these sections contain a proof of a theorem that we presented in Chapter 5. Before we go to the proofs, we first provide some additional (technical) information and notation.

For the proof of the convexity of the abstention curve (Theorem 5.2), we note that an ROC curve is convex if for all points (fpr_1, tpr_1), (fpr_2, tpr_2), and (fpr_3, tpr_3) such that $0 \leq fpr_1 < fpr_2 < fpr_3 \leq 1$, it holds that

$$tpr_2 \geq \frac{fpr_3 - fpr_2}{fpr_3 - fpr_1} tpr_1 + \frac{fpr_2 - fpr_1}{fpr_3 - fpr_1} tpr_3 \ , \tag{A.1}$$

or equivalently

$$\frac{tpr_2 - tpr_1}{fpr_2 - fpr_1} \geq \frac{tpr_3 - tpr_2}{fpr_3 - fpr_2} \ . \tag{A.2}$$

Also, for the proofs of the dominance relations between the original curve and the abstention curve (Theorems 5.3 to 5.5), we denote the points on the original curve by $(x, tpr(x))$. So, we introduce the function $tpr(x)$ for the highest true positive rate corresponding to a fixed false positive rate x. Analogously, the points $(x, tpr'(x))$ denote the points on the abstention curve. We recall that we may assume, without any restriction, that the original curves are convex.

For the proofs of the effectiveness of the ROC isometrics approach (Theorems 5.7 to 5.11), we assume for simplicity that $c = N/P$. We remember that the results generalize in a straightforward way to the case in which the skew ratio also incorporates a cost distribution. Finally, we note that after transformation to an abstaining classifier, a new skew ratio is obtained which we denote by $c' = c\ (1 - unr)/(1 - upr)$.

Proof of Theorem 5.1

The classifier represented by a point (fpr_i, tpr_i) between the points $(0,0)$ and (fpr_a, tpr_a) on the original curve classifies $tpr_i P$ instances correctly as positive and $fpr_i N$ instances incorrectly as positive. The classifier represented by the point (fpr'_i, tpr'_i) on the abstention curve classifies $tpr'_i(P - UP)$ instances correctly as positive and $fpr'_i(N - UN)$ instances incorrectly as positive. Since these instances are the same we have that

$$\begin{aligned} fpr_i N &= fpr'_i(N - UN) \text{ , and} \\ tpr_i P &= tpr'_i(P - UP) \text{ ,} \end{aligned}$$

and straightforward rewriting results in (5.4).

Analogously, the classifier represented by a point (fpr_i, tpr_i) between the points (fpr_b, tpr_b) and $(1,1)$ on the original curve classifies $(1 - fpr_i)N$ instances correctly as negative and $(1 - tpr_i)P$ instances incorrectly as negative. The classifier of the point (fpr'_i, tpr'_i) on the abstention curve classifies $(1 - fpr'_i)(N - UN)$ instances correctly as negative and $(1 - tpr'_i)(P - UP)$ instances incorrectly as negative. Since these instances are the same we have that

$$\begin{aligned} (1 - fpr_i)N &= (1 - fpr'_i)(N - UN) \text{ , and} \\ (1 - tpr_i)P &= (1 - tpr'_i)(P - UP) \text{ ,} \end{aligned}$$

and rewriting results in (5.5). □

Proof of Theorem 5.2

For the abstention curve to be convex, we need to show that either (A.1) or (A.2) holds for all points (fpr'_1, tpr'_1), (fpr'_2, tpr'_2), and (fpr'_3, tpr'_3) such that $0 \leq fpr'_1 < fpr'_2 < fpr'_3 \leq 1$. We split our proof into five different cases.

- **Case 1:** $0 \leq fpr'_1 < fpr'_2 < fpr'_3 \leq fpr_a$.
  The use of (5.4) shows that (A.2) for the abstention curve is equivalent to

  $$\frac{(tpr_2 - tpr_1)(1 - unr)}{(fpr_2 - fpr_1)(1 - upr)} \geq \frac{(tpr_3 - tpr_2)(1 - unr)}{(fpr_3 - fpr_2)(1 - upr)} \text{ ,}$$

  and this holds because of the convexity of the original curve.

- **Case 2:** $fpr'_b \leq fpr'_1 < fpr'_2 < fpr'_3 \leq 1$.
  The proof of this case is similar to that of case 1.

- **Case 3:** $0 \leq fpr'_1 < fpr'_2 = fpr'_a = fpr'_b < fpr'_3 \leq 1$.
  The analogue of (A.2) for the abstention curve is

  $$\frac{tpr'_a - tpr'_1}{fpr'_a - fpr'_1} \geq \frac{tpr'_3 - tpr'_b}{fpr'_3 - fpr'_b} \text{ ,} \tag{A.3}$$

and can be rewritten as

$$\frac{(tpr_a - tpr_1)(1-unr)}{(fpr_a - fpr_1)(1-upr)} \geq \frac{(tpr_3 - tpr_b)(1-unr)}{(fpr_3 - fpr_b)(1-upr)} \ , \tag{A.4}$$

using (5.4) and (5.5) for the left hand side and the right hand side of (A.3), respectively. Then we see that (A.4) holds because of the convexity of the original curve:

$$\frac{tpr_a - tpr_1}{fpr_a - fpr_1} \geq \frac{tpr_b - tpr_a}{fpr_b - fpr_a} \geq \frac{tpr_3 - tpr_b}{fpr_3 - fpr_b} \ .$$

- **Case 4:** $0 \leq fpr'_1 < fpr'_2 < fpr'_a = fpr'_b < fpr'_3 \leq 1$.
  We write analogues of (A.1) for the abstention curve. We know from case 1 that

$$tpr'_2 \geq \frac{fpr'_a - fpr'_2}{fpr'_a - fpr'_1} tpr'_1 + \frac{fpr'_2 - fpr'_1}{fpr'_a - fpr'_1} tpr'_a \ , \tag{A.5}$$

  and from case 3 that

$$tpr'_a \geq \frac{fpr'_3 - fpr'_a}{fpr'_3 - fpr'_2} tpr'_2 + \frac{fpr'_a - fpr'_2}{fpr'_3 - fpr'_2} tpr'_3 \ . \tag{A.6}$$

  The use of (A.6) to substitute tpr'_a in (A.5) results in

$$tpr'_2 \geq \frac{fpr'_3 - fpr'_2}{fpr'_3 - fpr'_1} tpr'_1 + \frac{fpr'_2 - fpr'_1}{fpr'_3 - fpr'_1} tpr'_3 \ .$$

- **Case 5:** $0 \leq fpr'_1 < fpr'_a = fpr'_b < fpr'_2 < fpr'_3 \leq 1$.
  The proof of this case is similar to that of case 4. □

Proof of Theorem 5.3

The abstention curve dominates the original one between the points $(0,0)$ and (fpr_a, tpr_a) when $tpr'(x) \geq tpr(x)$ for $0 \leq x \leq fpr_a$. The part of the original curve from the point $(0,0)$ to the point (fpr_a, tpr_a) is transformed into the abstention curve according to

$$(x, tpr(x)) \rightarrow \left(\frac{x}{1-unr}, \frac{tpr(x)}{1-upr} \right) \ .$$

It follows that

$$tpr' \left(\frac{x}{1-unr} \right) = \frac{tpr(x)}{1-upr} \ ,$$

or equivalently

$$tpr'(x) = \frac{tpr\left(x(1-unr)\right)}{1-upr} \ .$$

Now we see that $tpr'(x) \geq tpr(x)$ for $0 \leq x \leq fpr_a$ iff

$$tpr(x(1-unr)) \geq (1-upr)tpr(x) \ . \tag{A.7}$$

To proof (A.7) we use the fact that the original curve is convex between the points $(0,0)$ and $(x, tpr(x))$. Hence, (A.1) for $fpr_1 = 0$, $fpr_2 = x(1-unr)$, and $fpr_3 = x$ reads as follows:

$$tpr(x(1-unr)) \geq \frac{x - x(1-unr)}{x} 0 + \frac{x(1-unr)}{x} tpr(x) \ . \tag{A.8}$$

From a direct comparison between (A.7) and (A.8) we see that (A.7) is true if $unr \leq upr$. □

Proof of Theorem 5.4

The proof is similar to that of Theorem 5.3. □

Proof of Theorem 5.5

Note that $tpr(0) = tpr_0 > 0$. We start the proof with (A.7). Since the original curve is convex between the points $(0, tpr_0)$ and $(x, tpr(x))$ for $0 \leq x \leq fpr_a$, (A.1) for $fpr_1 = 0$, $fpr_2 = x(1-unr)$, and $fpr_3 = x$ results in

$$tpr(x(1-unr)) \geq unr\ tpr_0 + (1-unr)tpr(x) \ .$$

Thus, (A.7) holds iff

$$unr\ tpr_0 + (1-unr)tpr(x) \geq (1-upr)tpr(x) \ ,$$

or equivalently iff

$$tpr_0 \geq \left(1 - \frac{upr}{unr}\right) tpr(x) \ .$$

This holds when $unr \leq upr$ (cf. Theorem 5.3), and when both $unr > upr$ and $tpr(x) \leq \frac{1}{1-\frac{upr}{unr}} tpr_0$. The last inequality holds for the whole interval $0 \leq x \leq fpr_a$ iff $tpr_a \leq \frac{1}{1-\frac{upr}{unr}} tpr_0$.

From Theorem 5.4 and the result above it follows that the abstention curve dominates the original curve on the covered part if $tpr_0 > 0$, $unr > upr$, and if $tpr_a \leq \frac{1}{1-\frac{upr}{unr}} tpr_0$. □

Proof of Theorem 5.6

The proof is similar to that of Theorem 5.5. □

Proof of Theorem 5.7

The positive precisions in (fpr_a, tpr_a) and (fpr'_a, tpr'_a) are defined as follows:

$$prec_p^c\left(fpr_a, tpr_a\right) = \frac{tpr_a}{tpr_a + c\, fpr_a} \text{ , and} \tag{A.9}$$

$$prec_p^{c'}\left(fpr'_a, tpr'_a\right) = \frac{tpr'_a}{tpr'_a + c'\, fpr'_a} \text{ .} \tag{A.10}$$

Substitution of (5.4) in (A.10) results in (A.9). Analogously, we can use (5.5) to show that the negative precisions in (fpr_b, tpr_b) and (fpr'_b, tpr'_b) are equal. The theorem follows since $(fpr'_b, tpr'_b) = (fpr'_a, tpr'_a)$. □

Proof of Theorem 5.8

Since the positive precision and the negative precision in (fpr'_a, tpr'_a) are equal, we can define $a = prec_p^{c'} = prec_n^{c'}$ and rewrite the equations of the two variants of the precision metric to obtain

$$\begin{aligned} tpr'_a &= a\left(tpr'_a + c'\, fpr'_a\right) \text{ , and} \\ tnr'_a &= a\left(tnr'_a + \frac{1}{c'} fnr'_a\right) \text{ .} \end{aligned}$$

It follows that

$$tpr'_a + c'\, tnr'_a = a\left(tpr'_a + c'\, fpr'_a + c'\, tnr'_a + fnr'_a\right) \text{ ,}$$

or equivalently

$$a = \frac{tpr'_a + c'\, tnr'_a}{tpr'_a + c'\, fpr'_a + c'\, tnr'_a + fnr'_a} \text{ ,} \tag{A.11}$$

and this is the skew-sensitive version of accuracy with skew ratio c'. □

Proof of Theorem 5.9

The positive F-measures in (fpr_a, tpr_a) and (fpr'_a, tpr'_a) are defined as follows:

$$F_p^{c,\alpha}\left(fpr_a, tpr_a\right) = \frac{\left(1+\alpha^2\right) tpr_a}{\alpha^2 + tpr_a + c\, fpr_a} \text{ , and} \tag{A.12}$$

$$F_p^{c',\alpha}\left(fpr'_a, tpr'_a\right) = \frac{\left(1+\alpha^2\right) tpr'_a}{\alpha^2 + tpr'_a + c'\, fpr'_a} \text{ .} \tag{A.13}$$

Using (5.4) we can rewrite (A.13) to obtain

$$F_p^{c',\alpha}\left(fpr'_a, tpr'_a\right) = \frac{\left(1+\alpha^2\right) tpr_a}{\alpha^2(1-upr) + tpr_a + c\, fpr_a} \text{ .}$$

It follows that $F_p^{c',\alpha}\left(fpr'_a, tpr'_a\right) > F_p^{c,\alpha}\left(fpr_a, tpr_a\right)$ since $0 < upr < 1$. Analogously, we can use (5.5) to show that the negative F-measure in (fpr'_b, tpr'_b) is higher than the negative F-measure in (fpr_b, tpr_b). The theorem follows since we have by definition that $(fpr'_b, tpr'_b) = (fpr'_a, tpr'_a)$. □

Proof of Theorem 5.10

The positive m-estimates in (fpr_a, tpr_a) and (fpr'_a, tpr'_a) are defined as follows:

$$mest_p^{c,\hat{m}}\left(fpr_a, tpr_a\right) = \frac{tpr + \hat{m}}{tpr + c\,fpr + \hat{m}(1+c)} \text{ , and} \tag{A.14}$$

$$mest_p^{c',\hat{m}'}\left(fpr'_a, tpr'_a\right) = \frac{tpr' + \hat{m}'}{tpr' + c'\,fpr' + \hat{m}'(1+c')} \text{ .} \tag{A.15}$$

As explained in the main text, two cases are distinguished dependent on what we leave unchanged after transformation to an abstaining classifier.

- **Case 1:** m is not changed after transformation.
  In this case we have

  $$\hat{m}' = \frac{m}{P(1-upr) + N(1-unr)} \text{ .} \tag{A.16}$$

  Substitution of (5.4) and (A.16) in (A.15) followed by a straightforward calculation results in

  $$mest_p^{c',\hat{m}'}\left(fpr'_a, tpr'_a\right) = \frac{tpr + m\frac{1-upr}{P(1-upr)+N(1-unr)}}{tpr + c\,fpr + \hat{m}(1+c)} \text{ .} \tag{A.17}$$

  From a direct comparison between (A.14) and (A.17) it follows that the claim $mest_p^{c',\hat{m}'}\left(fpr'_a, tpr'_a\right) \geq mest_p^{c,\hat{m}}\left(fpr_a, tpr_a\right)$ holds iff

  $$\frac{1-upr}{P(1-upr) + N(1-unr)} \geq \frac{1}{P+N} \text{ .}$$

  This is true iff $upr \leq unr$.

- **Case 2:** $\hat{m}$ is not changed after transformation.
  In this case we have $\hat{m}' = \hat{m}$. Substitution of (5.4) in (A.15) followed by a straightforward calculation results in

  $$mest_p^{c',\hat{m}'}\left(fpr'_a, tpr'_a\right) = \frac{tpr + \hat{m}(1-upr)}{tpr + c\,fpr + \hat{m}(1-upr+c(1-unr))} \text{ .} \tag{A.18}$$

  From a direct comparison between (A.14) and (A.18) it follows that the claim $mest_p^{c',\hat{m}}\left(fpr'_a, tpr'_a\right) \geq mest_p^{c,\hat{m}}\left(fpr_a, tpr_a\right)$ holds iff

  $$\hat{m}(unr - upr) + (tpr_a\ unr - fpr_a\ upr) \geq 0 \text{ .}$$

  This is true if $upr \leq unr$ and $tpr_a \geq fpr_a$.

□

Proof of Theorem 5.11

The proof is similar to that of Theorem 5.10. □

Summary

Machine learning has been promoted many times as an important tool that can be used by law enforcement agencies to prevent crime and provide security to civilians. We fully agree and mention in Chapter 1 several examples of the important use of classifiers in the law enforcement field. Nonetheless, when we consider the implementations in practice, we may conclude that classifiers are only used for relatively easy tasks. We analyse why this is the case and present three domain-specific problems that should be alleviated or overcome. The key message is that conventional classifiers have difficulties to guarantee legally correct decisions and a correct treatment of civilians. Classifiers are not considered to be "reliable". Therefore, we have formulated the following problem statement: *To what extent can machine learning classifiers be used to increase the effectiveness and efficiency of law enforcement?* We note that, despite our motivation, the research presented in this thesis is also important for all other application fields.

After providing some background in Chapter 2, we start our research in Chapter 3 by studying RQ 1: *To what extent is the area under the ROC curve an effective performance metric for bipartite ranking when compared to its variants that consider the absolute values of the scores?* Assuming two possible class labels (positive and negative), we focus on ranking instances from most likely positive to most likely negative based on their scores, which are estimations of the probability that the instances are in fact positive instances. The ranking setting has large benefits for practical applications, especially in the field of law enforcement where predicting the most likely class label is not always sufficient. The area under the ROC curve, abbreviated AUC, has been widely-used to measure the ranking performance, but recently, it was conjectured to have a drawback since it does not consider the absolute values of the scores. Instead, it simply considers the sign of the difference in scores between pairs of positive and negative instances. It was, however, argued by several authors that smaller differences should be treated with more caution since they are less reliable than large differences. Several variants of the AUC estimate have been proposed for this reason. To answer RQ 1, we present a unified framework for the estimate and its variants. From our theoretical results, we may conclude that the variants are all biased and their variance can go in either direction. The net effect of the quality of the estimations is thus not clear. In addition, experiments with synthetic data show that the AUC is the most effective performance metric for model evaluation and selection. Our experimental analysis with benchmark data

sets confirms these findings: dependent on the model selection scenario, the AUC is competitive with its variants or it clearly outperforms the variants.

Subsequently, in Chapter 4, we are interested in RQ 2: *How can the AUC of an interpretable and comprehensible classifier, such as decision trees, be optimised?* Decision trees, trained in the usual way as a classifier, can be used for ranking by scoring an instance in terms of the frequency of positive training examples in the leaf to which the instance is assigned. However, several experimental results indicate that decision trees perform poorly in ranking instances, although the definition of the score is clearly reasonable. To answer RQ 2, we first replicate and extend previous empirical studies about approaches (which are not formally well-founded) to improve the AUC of decision trees. Analysing the results does not only improve the understanding of earlier approaches, but also corrects some implicit assumptions and conjectures that have been made by several authors. For example, the effect of Laplace correction was so far featured to an increased reliability in the sense that it considers scores of small leafs to be not as reliable as scores of large leafs. Our experiments, on the other hand, show that the benefit of Laplace correction comes from a reasonable local tie breaking effect that comes along with an increased number of distinct scores. From the theoretical results and accompanying simulation studies, we can confirm that the (empirical) AUC increases with the number of scores produced by the tree, at least when these scores are better than random approximations of the true conditional probabilities. Indeed, leaf-splitting seems to be tolerant toward estimation errors of probabilities for the positive class in the sense that only the ordering of scores is important. Even more important for practical purposes, we present a simple but very efficient and effective method for AUC-optimising decision trees. Our findings generalise in a straightforward way to other classifiers, as we illustrate with a weighted nearest neighbour classifier.

In Chapter 5 we study RQ 3: *Can we develop a feasible approach by which a classifier is constructed that guarantees a preset classification performance on each class?* Our answer to the research question is based on the concept of self-awareness in the following way. Instead of restricting the classifier to predict always one of the possible class labels, the classifier is given the option to say "I do not know" and consequently it refrains from classifying an instance (for which it is uncertain). The goal is then to design an approach for which the number of classification rejections is minimal, and yet, the desired performance values are guaranteed. Since we know *a priori* what we can expect from the classifier, we call it a reliable classifier. We provide an affirmative answer to the research question by means of introducing the ROC isometrics approach. This is an effective and generally applicable approach that has several advantages over other approaches that try to boost classification performance. A formal analysis was presented when performance is defined in terms of the following metrics: precision, accuracy, F-measure, and m-estimate. The implicit assumption of the approach is that the empirical ROC curve is an accurate estimate of the true curve. The validity of the approach is also tested using benchmark data sets. From the results, we may conclude that the ROC isometrics approach indeed results in classifiers that are reliable, at least, when sufficient instances are being processed.

So far, we have considered ranking and binary classification problems. A common strategy to solve a multi-class problem is to decompose it into a series of binary problems and to learn a set of corresponding base classifiers. The question is then how to aggregate the predictions of these base classifiers into a single final classification. Therefore, in Chapter 6 we study RQ 4: *Can we develop an aggregation strategy that is feasible in practice and shown to be optimal under reasonable conditions?* Using the setting of label ranking, we develop an aggregation strategy that takes the strength of the base classifiers into account. In this way, classifiers with high performance are seen as more reliable than classifiers with lower performance, and the aggregation strategy becomes less suspectible to (likely incorrect) outputs from the weak and unreliable classifiers. This strategy is called adaptive voting and shown to be optimal in the sense of yielding a maximum a posteriori prediction of the class label of a test instance. The conditions for this optimality are clearly stated and shown to be reasonable for actual implementation in practice. Next to this important contribution, we offer hitherto missing theoretical arguments in favour of the simple weighted voting, a strategy which has shown very good performance in practice. So far, weighted voting was considered to be ad hoc since no theoretical results concerning its performance existed. We show that weighted voting approximates the optimal adaptive voting prediction, and moreover, it has the advantage of being more robust when the model assumptions of adaptive voting are violated. From our experimental results with synthetic and real benchmark data sets, we can verify all these results in a consistent way.

Finally, in Chapter 7 we restate briefly our answers to the four research questions and formulate a conclusive answer to the problem statement. We may conclude that the research presented in this thesis results in a safer and more reliable use of machine learning classifiers in the field of law enforcement. The proposed approaches suggest a large step toward the implementation of intelligence led policing, eventually resulting in an increased effectiveness and efficiency of law enforcement. We end with several directions for future research to improve on the results of this thesis.

Samenvatting

Machine learning is vele malen naar voren gebracht als een belangrijk hulpmiddel dat kan worden gebruikt door de wetshandhavers om misdaden te voorkomen en de veiligheid van burgers te beschermen. Wij zijn het hiermee eens en geven in Hoofdstuk 1 verscheidene voorbeelden van belangrijke toepassingen van classificatiemodellen in het domein van de wetshandhaving. Niettemin, wanneer we kijken naar de implementaties in de praktijk, blijkt dat classificatiemodellen enkel gebruikt worden voor relatief eenvoudige taken. We analyseren waarom dit het geval is en introduceren drie domein-specifieke problemen die moeten worden voorkomen, of waarvan op zijn minst de effecten moeten worden gereduceerd. De hoofdboodschap is dat classificatiemodellen moeilijkheden hebben met het garanderen van juridisch correcte beslissingen en een correcte behandeling van burgers. Daarom worden de modellen beschouwd als niet "betrouwbaar". Om deze reden hebben we de volgende probleemstelling geformuleerd: *In welke mate kunnen machine learning classificatiemodellen worden gebruikt om de effectiviteit en efficiëntie van de rechtshandhaving te verbeteren?* We merken op dat, desondanks onze motivatie, het onderzoek voorgesteld in deze thesis ook belangrijk is voor alle andere toepassingsdomeinen.

Na de achtergrond besproken te hebben in Hoofdstuk 2, beginnen we ons onderzoek in Hoofdstuk 3 met het bestuderen van de eerste onderzoeksvraag (RQ, afgekort van het Engelse "research question"), RQ 1: *In welke mate is de oppervlakte onder de ROC-curve een effectieve maatstaf voor bipartite ordening wanneer het wordt vergeleken met haar varianten die de absolute waarden van de scores beschouwen?* Onder de aanname van twee mogelijke classificaties (positief en negatief) concentreren we ons op het ordenen van instanties van meest waarschijnlijk positief tot meest waarschijnlijk negatief. Deze ordening is gebaseerd op scores, wat schattingen zijn van de kans dat de instanties in feite positieve instanties zijn. Instanties ordenen heeft grote voordelen voor applicaties, in het bijzonder voor de rechtshandhaving waar het voorspellen van de meest waarschijnlijke classificatie niet altijd voldoende is. De oppervlakte onder de ROC-curve, of AUC in het kort (volgens het Engelse "area under the curve"), wordt bijna altijd gebruikt voor het meten van het ordeningsvermogen, maar recent kwam er kritiek omdat de metriek de absolute waarden van de scores niet beschouwt. Wat het wel doet, is het beschouwen van het teken van de verschillen in scores van paren bestaande uit positieve en negatieve instanties. Er werd echter geopperd door verschillende auteurs dat kleinere verschillen met meer voorzichtigheid moeten worden behandeld omdat zij minder betrouwbaar zijn dan

grote verschillen. Verschillende varianten van de AUC schattingen zijn daarom geïntroduceerd. Om RQ 1 te beantwoorden, introduceren we een verenigend kader voor de schatting van de AUC en haar varianten. Uit onze theoretische resultaten mogen we concluderen dat dat de varianten allemaal een *bias* hebben en hun variantie kan in beide richtingen evolueren. Het netto effect op de kwaliteit van de schatting is dus niet duidelijk. Verder hebben studies met synthetische data aangetoond dat de AUC inderdaad de meest effectieve metriek is voor modelevaluatie en -selectie. Onze experimenten met echte data bevestigen deze bevindingen.

Vervolgens leggen we ons in Hoofdstuk 4 toe op de beantwoording van RQ 2: *Hoe kan de AUC van een interpreteerbaar en begrijpbaar model, zoals beslissingsbomen, worden geoptimaliseerd?* Beslissingsbomen, geleerd in de gebruikelijke manier als classificatiemodellen, kunnen worden gebruikt voor ordening door de score van een instantie te definiëren als de frequentie van positieve trainingsvoorbeelden in het blad waaraan de instantie is toegewezen. Jammer genoeg hebben verschillende experimenten aangetoond dat beslissingsbomen niet goed werken voor ordening, ondanks het feit dat de definitie van de score realistisch is. Om RQ 2 te beantwoorden, gaan we eerst eerdere experimentele studies omtrent methoden (die niet formeel onderbouwd zijn) voor AUC-optimaliserende beslissingsbomen opnieuw testen, maar dan wel in meer detail. Een analyse van de resultaten leidt niet enkel tot een verbetering van ons begrip van deze methoden, maar ook tot een correctie van sommige impliciete aannames en beweringen die werden gemaakt door verschillende auteurs. Bijvoorbeeld, het effect van Laplace correctie werd tot nu toe toegeschreven aan een toegenomen betrouwbaarheid omdat de scores van kleine bladen worden beschouwd als minder nauwkeurig dan de scores van grote bladen. Onze experimenten, daarentegen, tonen aan dat het voordeel van Laplace correctie voortvloeit uit het feit dat het lokaal de banden breekt tussen instanties met eenzelfde score, en hierbij dus het aantal verschillende scores verhoogt. Uit onze theoretische resultaten en bijhorende simulatiestudies kunnen we bevestigen dat de (empirische) AUC toeneemt met het aantal scores dat een boom produceert, tenminste wanneer deze scores beter zijn dan willekeurige benaderingen van de werkelijke conditionele kansen. Het is immers zo dat het breken van instanties met dezelfde score tolerant is met betrekking tot schattingsfouten van de kansen omdat enkel de ordening van belang is. Nog meer van belang voor toepassingen in de praktijk is dat we een eenvoudige maar zeer efficiënte en effectieve methode introduceren voor AUC-optimaliserende beslissingsbomen. Onze bevindingen kunnen op een eenvoudige manier worden gegeneraliseerd naar andere classificatiemodellen.

In Hoofdstuk 5 bestuderen we RQ 3: *Kunnen we een methode ontwikkelen om een model te leveren dat een vooropgesteld classificatievermogen voor elke klasse kan garanderen?* Ons antwoord op de onderzoeksvraag is gebaseerd op het concept van zelfbewustzijn. In plaats van het classificatiemodel te verplichten om één van de mogelijke classificaties te voorspellen, geven we het de optie om te zeggen "Ik weet het niet" en bijgevolg onthoudt het model zich om de instantie (waarvoor het onzeker is) te classificeren. Het doel is dan het ontwikkelen van een methode waarbij het aantal onthoudingen minimaal is terwijl toch het vooropgestelde classificatievermogen wordt gegarandeerd. We noemen het resulterende model betrouwbaar omdat

we *a priori* weten wat we ervan kunnen verwachten. Een bevestigend antwoord op de onderzoeksvraag wordt gegeven door de introductie van de ROC-isometrieken methode. Dit is een effectieve en algemeen toepasbare methode die verscheidene voordelen heeft in vergelijking met verwant onderzoek. Een formele analyse wordt voorgesteld in de gevallen dat het classificatievermogen wordt gemeten door de metrieken: *precision*, *accuracy*, *F-measure*, en *m-estimate*. De impliciete aanname van de methode is dat de empirische ROC-curve een nauwkeurige schatting is van de werkelijke curve. De validiteit van de methode wordt dan ook uitgebreid getest. Uit de resultaten mogen we concluderen dat de ROC-isometrieken methode inderdaad betrouwbare modellen kan ontwikkelen, als ten minste voldoende instanties worden behandeld.

Tot nu toe hebben we ons beperkt tot ordening en binaire classificatieproblemen. In Hoofdstuk 6 verklaren we waarom multi-class problemen vaak worden opgelost door ze op te splitsen in een set van binaire classificatieproblemen. Vervolgens wordt een set van bijhorende classificatiemodellen geleerd. De vraag is dan hoe we de beslissingen van deze modellen moeten aggregeren tot een uiteindelijke classificatie. Om deze reden bestuderen we RQ 4: *Kunnen we een aggregatiestrategie ontwikkelen die bruikbaar is in de praktijk en waarvoor kan worden aangetoond dat het optimaal is onder realistische aannames?* Gebruikmakend van het formele kader van *label ranking* ontwikkelen we een aggregatiestrategie die de "kracht" van de modellen in beschouwing neemt. Op deze manier worden modellen met een groot classificatievermogen beschouwd als betrouwbaarder dan andere met een lager classificatievermogen. Bijgevolg wordt de strategie minder gevoelig voor (waarschijnlijk incorrecte) uitvoer van de zwakke en onbetrouwbare modellen. We noemen deze strategie *adaptive voting* en tonen aan dat het optimaal is omdat het een *maximum a posteriori* voorspelling oplevert van de classificatie van een test instantie. De condities voor deze optimaliteit worden duidelijk aangeduid en zijn realistisch genoeg voor een implementatie in de praktijk. Naast deze belangrijke contributie presenteren we ook theoretische argumenten voor het simpele *weighted voting*, een strategie die het heel goed doet in de praktijk. Tot nu toe werd deze strategie beschouwd als ad hoc omdat geen theoretische resultaten erover bestaan. We tonen aan dat *weighted voting* de voorspelling van de optimale *adaptive voting* benadert, en het heeft als extra voordeel dat het meer robuust is wanneer de aannames van *adaptive voting* niet gelden. Uit onze experimentele resultaten met synthetische en algemeen gebruikte dataverzamelingen kunnen we al onze bevindingen verifiëren en staven.

Tot slot, in Hoofdstuk 7 herhalen we beknopt onze antwoorden op de vier onderzoeksvragen en we formuleren een concluderend antwoord op de probleemstelling. We mogen concluderen dat het onderzoek gepresenteerd in deze thesis leidt tot een veiliger en betrouwbaarder gebruik van machine learning classificatiemodellen dan tot nu toe mogelijk was. De voorgestelde methoden suggereren een grote stap naar de implementatie van rechtshandhaving die wordt geleid door "intelligentie" verkregen met automatische methoden. We eindigen met verscheidene richtingen voor verder onderzoek om de resultaten van deze thesis nog te verbeteren.

Curriculum Vitae

Stijn Vanderlooy was born in Hasselt, Belgium, on September 17, 1983. He attended secondary school at the Sint-Jan Berchmansinstituut in Zonhoven from 1995 to 2001 (math-sciences education). From 2001 to 2004 he studied computer science at Hasselt University (Belgium) which collaborated with Maastricht University (The Netherlands). He received his master's degree cum laude. In 2003 and 2004 he also participated in the Initiële Academische Lerarenopleiding at Hasselt University and received his degree cum laude. This degree allows him to teach at high schools. In the summer of 2003 he participated in a summer course at Baylor University, Texas. Immediately after receiving his master's degree, he was appointed at the Institute for Knowledge and Agent Technology (currently known as Department of Knowledge Engineering) of Maastricht University to pursue a PhD. In the framework of the NWO ToKeN program (project number 634.000.435), he investigated approaches that are able to extend machine learning methods such that they can improve the efficiency and effectiveness of law enforcement agencies. The research results are however not restricted to this specific application and, in general, lead to a more safe and reliable use of classifiers than was possible so far. Besides this scientific work, he was involved with contract research and guidance of master students. He was also responsible to establish a new machine learning course with solution and exercise manual as well as a `MatLab` toolbox. In 2008 he visited the research group of Professor Eyke Hüllermeier in Marburg, Germany, where he worked on various topics including the area under the ROC curve, classifier calibration, and aggregation strategies for learning by pairwise comparison. Stijn Vanderlooy is an official referee for several respected international journals and conferences.

Publications

The investigations performed during my Ph.D. research resulted in publications dealing with different topics. So far not all of them have been referenced. Therefore, a full list of publications is presented below. The list is divided into three parts, namely: journal articles, conference and workshop proceedings, and technical reports.

Journal articles

1. Ilkka Havukkala and Stijn Vanderlooy (2007). On the Reliable Identification of Plant Sequences Containing a Polyadenylation Site. *Journal of Computational Biology*, Vol. 14, No. 9, pp. 1229-1245.

2. Stijn Vanderlooy, Joop Verbeek, and Jaap van den Herik (2007). Towards Privacy-Preserving Data Mining in Law Enforcement. *Journal of International Commercial Law and Technology*, Vol. 2, No. 4, pp. 202-210.

3. Stijn Vanderlooy and Eyke Hüllermeier (2008). A Critical Analysis of Variants of the AUC. *Machine Learning*, Vol. 72, No. 3, pp. 247-262.

4. Benjamin Torben-Nielsen, Stijn Vanderlooy, and Eric Postma (2008). Non-Parametric Algorithmic Generation of Neuronal Morphologies. *Neuroinformatics*, Vol. 6, No. 4, pp. 257-277.

5. Stijn Vanderlooy, Ida Sprinkhuizen-Kuyper, Evgueni Smirnov, and Jaap van den Herik (2009). The ROC Isometrics Approach to Construct Reliable Classifiers. *Intelligent Data Analysis*, Vol. 13, No. 1, pp. 3-37.

6. Eyke Hüllermeier and Stijn Vanderlooy (2009). Why Fuzzy Decision Trees are Good Rankers. *IEEE Transactions on Fuzzy Systems*, Accepted.

7. Eyke Hüllermeier and Stijn Vanderlooy (2009). Combining Predictions in Pairwise Classification: An Optimal Adaptive Voting Strategy and Its Relation to Weighted Voting. *Pattern Recognition*, Under revision.

Conference and workshop proceedings

8. Stijn Vanderlooy, Joop Verbeek, and Jaap van den Herik (2005). Kunstmatige Intelligentie en De Wet Politiegegevens. In Wim Huisman, Martin Moerings, and Guido Suurmond, editors, *Veiligheid en Recht: Nieuwe Doelwitten, Nieuwe Strategieën*, pp. 255-266, Leiden, the Netherlands, November 25. Boom Juridische Uitgevers (in Dutch).

9. Stijn Vanderlooy, Ida Sprinkhuizen-Kuyper, and Evgueni Smirnov (2006). Reliable Classifiers in ROC Space. In Ivan Saeys, Elena Tsiporkova, Bernard De Baets, and Yves van de Peer, editors, *Proceedings of the 15th Annual Machine Learning Conference of Belgium and the Netherlands*, pp. 113-120, Ghent, Belgium, May 11-22.

10. Stijn Vanderlooy, Ida Sprinkhuizen-Kuyper, and Evgueni Smirnov (2006). An Analysis of Reliable Classifiers Through ROC Isometrics. In Nicolas Lachiche, César Ferri, and Sofus Macskassy, editors, *Proceedings of the ICML 2006 Workshop on ROC Analysis*, pp. 55-62, Pittsburgh, USA, June 29.

11. Evgueni Smirnov, Ida Sprinkhuizen-Kuyper, Georgi Nalbantov, and Stijn Vanderlooy (2006). Version Space Support Vector Machines. In Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso, editors, *Proceedings of the 17th European Conference on Artificial Intelligence*, pp. 809-810, Riva del Garda, Spain, August 28 - September 1. IOS Press.

12. Evgueni Smirnov, Stijn Vanderlooy, and Ida Sprinkhuizen-Kuyper (2006). Meta-Typicalness Approach to Reliable Classification. In Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso, editors, *Proceedings of the 17th European Conference on Artificial Intelligence*, pp. 810-811, Riva del Garda, Spain, August 28 - September 1. IOS Press.

13. Stijn Vanderlooy, Eric Postma, Karl Tuyls, and Ida Sprinkhuizen-Kuyper (2006). Reliable Instance Classifications in Law Enforcement. In Pierre-Yves Schobbens, Wim Vanhoof, and Gabriel Schwanen, editors, *Proceedings of the 18th Benelux Conference on Artificial Intelligence*, pp. 323-330, Namur, Belgium, October 5-6.

14. Stijn Vanderlooy, Joop Verbeek, and Jaap van den Herik. Towards Privacy-Preserving Data Mining in Law Enforcement (2007). In Sylvia Kierkegaard, editor, *Proceedings of the International Law and Trade Conference*, pp. 384-392, Istanbul, Turkey, May 10-11. Ankara Bar Association Press.

15. Stijn Vanderlooy, Laurens van der Maaten, and Ida Sprinkhuizen-Kuyper (2007). Off-line Learning with Transductive Confidence Machines: An Empirical Evaluation. In Petra Perner, editor, *Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition*, pp. 310-323, Leipzig, Germany, July 18-20. Springer.

16. Stijn Vanderlooy and Ida Sprinkhuizen-Kuyper (2007). A Comparison of Two Approaches to Classify with Guaranteed Performance. In Joost Kok, Jacek

Koronacki, Ramon López de Mántaras, Stan Matwin, Dunja Mladenic and Andrzej Skowron, editors, *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp. 288-299, Warsaw, Poland, September 17-21. Springer.

17. Stijn Vanderlooy and Eyke Hüllermeier (2008). A Critical Analysis of Variants of the AUC. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *Proceedings of the 19th European Conference on Machine Learning*, p. 13, Antwerp, Belgium, September 15-19. Springer.

18. Eyke Hüllermeier and Stijn Vanderlooy (2008). Weighted Voting as Approximate MAP Prediction in Pairwise Classification. In Joachim Baumeister and Martin Atzmüller, editors, *Proceedings of the LWA Workshop on Knowledge Discovery, Data Mining, and Machine Learning*, pp. 34-41, Würzburg, Germany, October 6-8. University of Würzburg.

Technical Reports

19. Stijn Vanderlooy and Ida Sprinkhuizen-Kuyper (2007). An Overview of Algorithmic Randomness and its Application to Reliable Instance Classification. Technical Report MICC-IKAT 07-02, Maastricht University, the Netherlands.

20. Stijn Vanderlooy, Laurens van der Maaten, and Ida Sprinkhuizen-Kuyper (2007). Off-Line Learning with Transductive Confidence Machines: an Empirical Evaluation. Technical Report MICC-IKAT 07-03, Maastricht University, the Netherlands.

21. Eyke Hüllermeier and Stijn Vanderlooy (2008). An Empirical and Formal Analysis of Decision Trees for Ranking. Technical Report Computer Science Series 56, Marburg University, Germany.

22. Stijn Vanderlooy (2008). Matlab Toolbox for Machine Learning. Technical Report MICC 08-03, Maastricht University, the Netherlands.

SIKS Dissertation Series

1998

1 Johan van den Akker (CWI[1]) *DEGAS – An Active, Temporal Database of Autonomous Objects*

2 Floris Wiesman (UM) *Information Retrieval by Graphically Browsing Meta-Information*

3 Ans Steuten (TUD) *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*

4 Dennis Breuker (UM) *Memory versus Search in Games*

5 Eduard W. Oskamp (RUL) *Computerondersteuning bij Straftoemeting*

1999

1 Mark Sloof (VU) *Physiology of Quality Change Modelling; Automated Modelling of Quality Change of Agricultural Products*

2 Rob Potharst (EUR) *Classification using Decision Trees and Neural Nets*

3 Don Beal (UM) *The Nature of Minimax Search*

4 Jacques Penders (UM) *The Practical Art of Moving Physical Objects*

5 Aldo de Moor (KUB) *Empowering Communities: a Method for the Legitimate User-Driven Specification of Network Information Systems*

6 Niek J.E. Wijngaards (VU) *Re-Design of Compositional Systems*

7 David Spelt (UT) *Verification Support for Object Database Design*

8 Jacques H.J. Lenting (UM) *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation*

2000

1 Frank Niessink (VU) *Perspectives on Improving Software Maintenance*

2 Koen Holtman (TU/e) *Prototyping of CMS Storage Management*

3 Carolien M.T. Metselaar (UvA) *Sociaal-organisatorische Gevolgen van Kennistechnologie; een Procesbenadering en Actorperspectief*

4 Geert de Haan (VU) *ETAG, A Formal Model of Competence Knowledge for User Interface Design*

5 Ruud van der Pol (UM) *Knowledge-Based Query Formulation in Information Retrieval*

6 Rogier van Eijk (UU) *Programming Languages for Agent Communication*

7 Niels Peek (UU) *Decision-Theoretic Planning of Clinical Patient Management*

8 Veerle Coupé (EUR) *Sensitivity Analyis of Decision-Theoretic Networks*

[1] Abbreviations: SIKS – Dutch Research School for Information and Knowledge Systems; CWI – Centrum voor Wiskunde en Informatica, Amsterdam; EUR – Erasmus Universiteit, Rotterdam; KUB – Katholieke Universiteit Brabant, Tilburg; KUN – Katholieke Universiteit Nijmegen; RUL – Rijksuniversiteit Leiden; RUN – Radboud Universiteit Nijmegen; TUD – Technische Universiteit Delft; TU/e – Technische Universiteit Eindhoven; UL – Universiteit Leiden; UM – Universiteit Maastricht; UT – Universiteit Twente, Enschede; UU – Universiteit Utrecht; UvA – Universiteit van Amsterdam; UvT – Universiteit van Tilburg; VU – Vrije Universiteit, Amsterdam; RUN – Radboud Universiteit Nijmegen.

9 Florian Waas (CWI) *Principles of Probabilistic Query Optimization*

10 Niels Nes (CWI) *Image Database Management System Design Considerations, Algorithms and Architecture*

11 Jonas Karlsson (CWI) *Scalable Distributed Data Structures for Database Management*

2001

1 Silja Renooij (UU) *Qualitative Approaches to Quantifying Probabilistic Networks*

2 Koen Hindriks (UU) *Agent Programming Languages: Programming with Mental Models*

3 Maarten van Someren (UvA) *Learning as Problem Solving*

4 Evgueni Smirnov (UM) *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*

5 Jacco van Ossenbruggen (VU) *Processing Structured Hypermedia: a Matter of Style*

6 Martijn van Welie (VU) *Task-Based User Interface Design*

7 Bastiaan Schonhage (VU) *Diva: Architectural Perspectives on Information Visualization*

8 Pascal van Eck (VU) *A Compositional Semantic Structure for Multi-Agent Systems Dynamics*

9 Pieter Jan 't Hoen (RUL) *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*

10 Maarten Sierhuis (UvA) *Modeling and Simulating Work Practice; BRAHMS: a Multiagent Modeling and Simulation Language for Work Practice Analysis and Design*

11 Tom M. van Engers (VU) *Knowledge Management: The Role of Mental Models in Business Systems Design*

2002

1 Nico Lassing (VU) *Architecture-Level Modifiability Analysis*

2 Roelof van Zwol (UT) *Modelling and Searching Web-based Document Collections*

3 Henk Ernst Blok (UT) *Database Optimization Aspects for Information Retrieval*

4 Juan Roberto Castelo Valdueza (UU) *The Discrete Acyclic Digraph Markov Model in Data Mining*

5 Radu Serban (VU) *The Private Cyberspace Modeling Electronic Environments Inhabited by Privacy-Concerned Agents*

6 Laurens Mommers (UL) *Applied Legal Epistemology; Building a Knowledge-based Ontology of the Legal Domain*

7 Peter Boncz (CWI) *Monet: a Next-Generation DBMS Kernel For Query-Intensive Applications*

8 Jaap Gordijn (VU) *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*

9 Willem-Jan van den Heuvel (KUB) *Integrating Modern Business Applications with Objectified Legacy Systems*

10 Brian Sheppard (UM) *Towards Perfect Play of Scrabble*

11 Wouter C.A. Wijngaards (VU) *Agent Based Modelling of Dynamics: Biological and Organisational Applications*

12 Albrecht Schmidt (UvA) *Processing XML in Database Systems*

13 Hongjing Wu (TU/e) *A Reference Architecture for Adaptive Hypermedia Applications*

14 Wieke de Vries (UU) *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*

15 Rik Eshuis (UT) *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*

16 Pieter van Langen (VU) *The Anatomy of Design: Foundations, Models and Applications*

17 Stefan Manegold (UvA) *Understanding, Modeling, and Improving Main-Memory Database Performance*

2003

1 Heiner Stuckenschmidt (VU) *Ontology-Based Information Sharing in Weakly Structured Environments*

2 Jan Broersen (VU) *Modal Action Logics for Reasoning About Reactive Systems*

3 Martijn Schuemie (TUD) *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*

4 Petkovic (UT) *Content-Based Video Retrieval Supported by Database Technology*

5 Jos Lehmann (UvA) *Causation in Artificial Intelligence and Law – A Modelling Approach*

6 Boris van Schooten (UT) *Development and Specification of Virtual Environments*

7 Machiel Jansen (UvA) *Formal Explorations of Knowledge Intensive Tasks*

8 Yong-Ping Ran (UM) *Repair-Based Scheduling*

9 Rens Kortmann (UM) *The Resolution of Visually Guided Behaviour*

10 Andreas Lincke (UT) *Electronic Business Negotiation: Some Experimental Studies on the Interaction between Medium, Innovation Context and Cult*

11 Simon Keizer (UT) *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*

12 Roeland Ordelman (UT) *Dutch Speech Recognition in Multimedia Information Retrieval*

13 Jeroen Donkers (UM) *Nosce Hostem – Searching with Opponent Models*

14 Stijn Hoppenbrouwers (KUN) *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*

15 Mathijs de Weerdt (TUD) *Plan Merging in Multi-Agent Systems*

16 Menzo Windhouwer (CWI) *Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouse*

17 David Jansen (UT) *Extensions of Statecharts with Probability, Time, and Stochastic Timing*

18 Levente Kocsis (UM) *Learning Search Decisions*

2004

1 Virginia Dignum (UU) *A Model for Organizational Interaction: Based on Agents, Founded in Logic*

2 Lai Xu (UvT) *Monitoring Multi-party Contracts for E-business*

3 Perry Groot (VU) *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*

4 Chris van Aart (UvA) *Organizational Principles for Multi-Agent Architectures*

5 Viara Popova (EUR) *Knowledge Discovery and Monotonicity*

6 Bart-Jan Hommes (TUD) *The Evaluation of Business Process Modeling Techniques*

7 Elise Boltjes (UM) *Voorbeeld$_{IG}$ Onderwijs; Voorbeeldgestuurd Onderwijs, een Opstap naar Abstract Denken, vooral voor Meisjes*

8 Joop Verbeek (UM) *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale Politiële Gegevensuitwisseling en Digitale Expertise*

9 Martin Caminada (VU) *For the Sake of the Argument; Explorations into Argument-based Reasoning*

10 Suzanne Kabel (UvA) *Knowledge-rich Indexing of Learning-objects*

11 Michel Klein (VU) *Change Management for Distributed Ontologies*

12 The Duy Bui (UT) *Creating Emotions and Facial Expressions for Embodied Agents*

13 Wojciech Jamroga (UT) *Using Multiple Models of Reality: On Agents who Know how to Play*

14 Paul Harrenstein (UU) *Logic in Conflict. Logical Explorations in Strategic Equilibrium*

15 Arno Knobbe (UU) *Multi-Relational Data Mining*

16 Federico Divina (VU) *Hybrid Genetic Relational Search for Inductive Learning*

17 Mark Winands (UM) *Informed Search in Complex Games*

18 Vania Bessa Machado (UvA) *Supporting the Construction of Qualitative Knowledge Models*

19 Thijs Westerveld (UT) *Using Generative Probabilistic Models for Multimedia Retrieval*

20 Madelon Evers (Nyenrode) *Learning from Design: Facilitating Multidisciplinary Design Teams*

2005

1 Floor Verdenius (UvA) *Methodological Aspects of Designing Induction-Based Applications*

2 Erik van der Werf (UM) *AI Techniques for the Game of Go*

3 Franc Grootjen (RUN) *A Pragmatic Approach to the Conceptualisation of Language*

4 Nirvana Meratnia (UT) *Towards Database Support for Moving Object Data*

5 Gabriel Infante-Lopez (UvA) *Two-Level Probabilistic Grammars for Natural Language Parsing*

6 Pieter Spronck (UM) *Adaptive Game AI*

7 Flavius Frasincar (TUE) *Hypermedia Presentation Generation for Semantic Web Information Systems*

8 Richard Vdovjak (TUE) *A Model-driven Approach for Building Distributed Ontology-based Web Applications*

9 Jeen Broekstra (VU) *Storage, Querying and Inferencing for Semantic Web Languages*

10 Anders Bouwer (UVA) *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*

11 Elth Ogston (VU) *Agent Based Matchmaking and Clustering - A Decentralized Approach to Search*

12 Csaba Boer (EUR) *Distributed Simulation in Industry*

13 Fred Hamburg (UL) *Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen*

14 Borys Omelayenko (VU) *Web-Service configuration on the Semantic Web; Exploring how Semantics meets Pragmatics*

15 Tibor Bosse (VU) *Analysis of the Dynamics of Cognitive Processes*

16 Joris Graaumans (UU) *Usability of XML Query Languages*

17 Boris Shishkov (TUD) *Software Specification Based on Re-usable Business Components*

18 Danielle Sent (UU) *Test-selection Strategies for Probabilistic Networks*

19 Michel van Dartel (UM) *Situated Representation*

20 Cristina Coteanu (UL) *Cyber Consumer Law, State of the Art and Perspectives*

21 Wijnand Derks (UT) *Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics*

2006

1 Samuil Angelov (TUE) *Foundations of B2B Electronic Contracting*

2 Cristina Chisalita (VU) *Contextual Issues in the Design and Use of Information Technology in Organizations*

3 Noor Christoph (UVA) *The Role of Metacognitive Skills in Learning to Solve Problems*

4 Marta Sabou (VU) *Building Web Service Ontologies*

5 Cees Pierik (UU) *Validation Techniques for Object-Oriented Proof Outlines*

6 Ziv Baida (VU) *Software-aided Service Bundling – Intelligent Methods & Tools for Graphical Service Modeling*

7 Marko Smiljanic (UT) *XML Schema Matching – Balancing Efficiency and Effectiveness by means of Clustering*

8 Eelco Herder (UT) *Forward, Back and Home Again – Analyzing User Behavior on the Web*

9 Mohamed Wahdan (UM) *Automatic Formulation of the Auditor's Opinion*

10 Ronny Siebes (VU) *Semantic Routing in Peer-to-Peer Systems*

11 Joeri van Ruth (UT) *Flattening Queries over Nested Data Types*

12 Bert Bongers (VU) *Interactivation – Towards an E-cology of People, our Technological Environment, and the Arts*

13 Henk-Jan Lebbink (UU) *Dialogue and Decision Games for Information Exchanging Agents*

14 Johan Hoorn (VU) *Software Requirements: Update, Upgrade, Redesign – towards a Theory of Requirements Change*

15 Rainer Malik (UU) *CONAN: Text Mining in the Biomedical Domain*

16 Carsten Riggelsen (UU) *Approximation Methods for Efficient Learning of Bayesian Networks*

17 Stacey Nagata (UU) *User Assistance for Multitasking with Interruptions on a Mobile Device*

18 Valentin Zhizhkun (UVA) *Graph Transformation for Natural Language Processing*

19 Birna van Riemsdijk (UU) *Cognitive Agent Programming: A Semantic Approach*

20 Marina Velikova (UvT) *Monotone Models for Prediction in Data Mining*

21 Bas van Gils (RUN) *Aptness on the Web*

22 Paul de Vrieze (RUN) *Fundaments of Adaptive Personalisation*

23 Ion Juvina (UU) *Development of Cognitive Model for Navigating on the Web*

24 Laura Hollink (VU) *Semantic Annotation for Retrieval of Visual Resources*

25 Madalina Drugan (UU) *Conditional log-likelihood MDL and Evolutionary MCMC*

26 Vojkan Mihajlović (UT) *Score Region Algebra: a Flexible Framework for Structured Information Retrieval*

27 Stefano Bocconi (CWI) *Vox Populi: Generating Video Documentaries from Semantically Annotated Media Repositories*

28 Borkur Sigurbjornsson (UVA) *Focused Information Access using XML Element Retrieval*

2007

1 Kees Leune (UvT) *Access Control and Service-Oriented Architectures*

2 Wouter Teepe (RUG) *Reconciling Information Exchange and Confidentiality: a Formal Approach*

3 Peter Mika (VU) *Social Networks and the Semantic Web*

4 Jurriaan van Diggelen (UU) *Achieving Semantic Interoperability in Multi-agent Systems: a Dialogue-based Approach*

5 Bart Schermer (UL) *Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance*

6 Gilad Mishne (UVA) *Applied Text Analytics for Blogs*

7 Natasa Jovanovic' (UT) *To Whom It May Concern – Addressee Identification in Face-to-Face Meetings*

8 Mark Hoogendoorn (VU) *Modeling of Change in Multi-Agent Organizations*

9 David Mobach (VU) *Agent-Based Mediated Service Negotiation*

10 Huib Aldewereld (UU) *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*

11 Natalia Stash (TUE) *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*

12 Marcel van Gerven (RUN) *Bayesian Networks for Clinical Decision Support: a Rational Approach to Dynamic Decision-Making under Uncertainty*

13 Rutger Rienks (UT) *Meetings in Smart Environments; Implications of Progressing Technology*

14 Niek Bergboer (UM) *Context-Based Image Analysis*

15 Joyca Lacroix (UM) *NIM: a Situated Computational Memory Model*

16 Davide Grossi (UU) *Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems*

17 Theodore Charitos (UU) *Reasoning with Dynamic Networks in Practice*

18 Bart Orriens (UvT) *On the Development and Management of Adaptive Business Collaborations*

19 David Levy (UM) *Intimate Relationships with Artificial Partners*

20 Slinger Jansen (UU) *Customer Configuration Updating in a Software Supply Network*

21 Karianne Vermaas (UU) *Fast Diffusion and Broadening Use: a Research on Residential Adoption and Usage of Broadband Internet in the Netherlands between 2001 and 2005*

22 Zlatko Zlatev (UT) *Goal-oriented Design of Value and Process Models from Patterns*

23 Peter Barna (TUE) *Specification of Application Logic in Web Information Systems*

24 Georgina Ramírez Camps (CWI) *Structural Features in XML Retrieval*

25 Joost Schalken (VU) *Empirical Investigations in Software Process Improvement*

2008

1 Katalin Boer-Sorbán (EUR) *Agent-Based Simulation of Financial Markets: a Modular, Continuous-time Approach*

2 Alexei Sharpanskykh (VU) *On Computer-Aided Methods for Modeling and Analysis of Organizations*

3 Vera Hollink (UVA) *Optimizing Hierarchical Menus: a Usage-based Approach*

4 Ander de Keijzer (UT) *Management of Uncertain Data – towards Unattended Integration*

5 Bela Mutschler (UT) *Modeling and Simulating Causal Dependencies on Process-aware Information Systems from a Cost Perspective*

6 Arjen Hommersom (RUN) *On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective*

7 Peter van Rosmalen (OU) *Supporting the Tutor in the Design and Support of Adaptive E-learning*

8 Janneke Bolt (UU) *Bayesian Networks: Aspects of Approximate Inference*

9 Christof van Nimwegen (UU) *The Paradox of the Guided User: Assistance can be Counter-effective*

10 Wauter Bosma (UT) *Discourse Oriented Summarization*

11 Vera Kartseva (VU) *Designing Controls for Network Organizations: a Value-Based Approach*

12 Jozsef Farkas (RUN) *A Semiotically Oriented Cognitive Model of Knowledge Representation*

13 Caterina Carraciolo (UVA) *Topic Driven Access to Scientific Handbooks*

14 Arthur van Bunningen (UT) *Context-Aware Querying; Better Answers with Less Effort*

15 Martijn van Otterlo (UT) *The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains*

16 Henriette van Vugt (VU) *Embodied Agents from a User's Perspective*

17 Martin Op 't Land (TUD) *Applying Architecture and Ontology to the Splitting and Allying of Enterprises*

18 Guido de Croon (UM) *Adaptive Active Vision*

19 Henning Rode (UT) *From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search*

20 Rex Arendsen (VA) *Geen Bericht, Goed Bericht. Een Onderzoek naar de Effecten van de Introductie van Elektronisch Berichtenverkeer met de Overheid op de Administratieve Lasten van Bedrijven*

21 Krisztian Balog (UVA) *People Search in the Enterprise*

22 Henk Koning (UU) *Communication of IT-Architecture*

23 Stefan Visscher (UU) *Bayesian Network Models for the Management of Ventilator-associated Pneumonia*

24 Zharko Aleksovski (VU) *Using Background Knowledge in Ontology Matching*

25 Geert Jonker (UU) *Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency*

26 Marijn Huijbregts (UT) *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled*

27 Hubert Vogten (OU) *Design and Implementation Strategies for IMS Learning Design*

28 Ildiko Flesch (RUN) *On the Use of Independence Relations in Bayesian Networks*

29 Dennis Reidsma (UT) *Annotations and Subjective Machines – Of Annotators, Embodied Agents, Users, and Other Humans*

30 Wouter van Atteveldt (VU) *Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content*

31 Loes Braun (UM) *Pro-Active Medical Information Retrieval*

32 Trung H. Bui (UT) *Toward Affective Dialogue Management using Partially Observable Markov Decision Processes*

33 Frank Terpstra (UVA) *Scientific Workflow Design – Theoretical and Practical Issues*

34 Jeroen de Knijf (UU) *Studies in Frequent Tree Mining*

35 Ben Torben Nielsen (UvT) *Dendritic Morphologies: Function Shapes Structure*

2009

1 Rasa Jurgelenaite (RUN) *Symmetric Causal Independence Models*

2 Willem Robert van Hage (VU) *Evaluating Ontology-Alignment Techniques*

3 Hans Stol (UvT) *A Framework for Evidence-based Policy Making Using IT*

4 Josephine Nabukenya (RUN) *Improving the Quality of Organisational Policy Making using Collaboration Engineering*

5 Sietse Overbeek (RUN) *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality*

6 Muhammad Subianto (UU) *Understanding Classification*

7 Ronald Poppe (UT) *Discriminative Vision-Based Recovery and Recognition of Human Motion*

8 Volker Nannen (VU) *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*

9 Benjamin Kanagwa (RUN) *Design, Discovery and Construction of Service-oriented Systems*

10 Jan Wielemaker (UVA) *Logic programming for knowledge-intensive interactive applications*

11 Alexander Boer (UVA) *Legal Theory, Sources of Law & the Semantic Web*

12 Peter Massuthe (TUE, Humboldt-Universität zu Berlin) *Operating Guidelines for Services*

13 Steven de Jong (UM) *Fairness in Multi-Agent Systems*

14 Maksym Korotkiy (VU) *From Ontology-Enabled Services to Service-Enabled Ontologies*

15 Rinke Hoekstra (UVA) *Ontology Representation - Design Patterns and Ontologies that Make Sense*

16 Fritz Reul (UvT) *New Architectures in Computer Chess*

17 Laurens van der Maaten (UvT) *Feature Extraction from Visual Data*

18 Fabian Groffen (CWI) *Armada, An Evolving Database System*

19 Valentin Robu (CWI) *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*

20 Bob van der Vecht (UU) *Armada, An Evolving Database System*

21 Stijn Vanderlooy (UM) *Ranking and Reliable Classification*